Tesis Doctoral

Reconocimiento de patrones utilizando técnicas estadísticas y conexionistas aplicadas a la clasificación de dígitos manuscritos

Seijas, Leticia María

2011

UNIVERSIDAD DE BUENOS AIRES

Facultad de Ciencias Exactas y Naturales

Departamento de Computación

Reconocimiento de patrones utilizando técnicas estadísticas y conexionistas aplicadas a la clasificación de dígitos manuscritos

Tesis presentada para optar al título de
Doctor de la Universidad de Buenos Aires
en el área Ciencias de la Computación

Leticia María Seijas

Director: Dr. Enrique Carlos Segura

Consejera de estudios: Dra. Ana Ruedin

Buenos Aires 2011

Pattern Recognition using Statistical Techniques and Neural Networks: Application to Handwritten Digit Classification

Abstract

Pattern Recognition is the study of how machines can observe the environment, learn to distinguish patterns of interest from their background, and make sound and reasonable decisions about the categories of the patterns. The best pattern recognizers in most instances are humans, yet we do not understand how humans recognize patterns.

Optical character recognition (OCR) is one of the most traditional topics in the context of Pattern Recognition that includes as a key issue the automatic recognition of handwritten characters. The subject has many interesting applications, such as automatic recognition of postal codes, recognition of amounts in banking checks and automatic processing of application forms. Handwritten numeral classification is a difficult task because of the wide variety of styles, strokes and orientations of digit samples. One of the main difficulties lies in the fact that the intra-class variance is high, due to the different forms associated with the same pattern, because of the particular writing style of each individual. Many models have been proposed to deal with this problem, but none of them has succeeded in obtaining levels of response comparable to human ones.

This thesis presents a pattern recognition system that is able to detect ambiguous patterns and explain its answers using a Bayesian strategy which is the main contribution of this work. The recogniser is composed of two levels. The first one is formed by a collection of independent classifiers, each one specialised in a different feature extracted from the input pattern. The second level consists of an analyzing module in charge of defining and explaining the output of the system. This module is integrated by the following elements: the table of reliability and two parameters adjustable while running the system.

The system has been applied to the off-line recognition of handwritten digits. Descriptors based on the CDF 9/7 wavelet transform and Principal Component Analysis are proposed in order to reduce the size of the input pattern while increasing the quality of its representation. Strategies for selecting classifiers for the system are also proposed.

The experiments were carried out on the MNIST and CENPARMI handwritten digit databases, which are generally accepted as standards in most of the literature in the field. Recognition rates obtained are comparable with results from representative work, reaching 97.40 and 99.32 % for CENPARMI and MNIST databases respectively.

Keywords: Pattern Recognition, handwritten digit classification, ambiguous pattern, neural networks, support vector machines, bayesian statistics.

Reconocimiento de Patrones utilizando Técnicas Estadísticas y Conexionistas aplicadas a la Clasificación de Dígitos Manuscritos

Resumen

El Reconocimiento de Patrones es el estudio de cómo las máquinas pueden observar el ambiente o entorno, aprender a distinguir patrones de interés a partir de la experiencia, y tomar decisiones razonables con respecto a las categorías a las que pertenecen dichos patrones. El mejor reconocedor de patrones conocido hasta ahora es el ser humano, no sabiéndose a ciencia cierta cuál es el proceso mediante el cual los humanos realizamos esta tarea. El Reconocimiento Óptico de Caracteres (OCR) es uno de los tópicos más antiguos dentro del Reconocimiento de Patrones y una de las áreas de investigación más importante y activa, que en la actualidad presenta desafíos: la precisión en el reconocimiento asociada tanto a caracteres impresos en una imagen degradada o a caracteres manuscritos es aún insuficiente, existiendo errores en el reconocimiento. El Reconocimiento de Dígitos Manuscritos es un tema destacado dentro de OCR, por las aplicaciones relacionadas, como el procesamiento automático de cheques bancarios, la clasificación de correo en base a la lectura de códigos postales, la lectura automática de formularios y documentos con escritura manuscrita, dispositivos de lectura para ciegos, reconocimiento de escritura en computadoras manuales PDA, y porque constituye un problema modelo que incluye desafíos comunes con otros tópicos. Por esta razón, es tomado como referencia para la aplicación y testeo de nuevas teorías y algoritmos del área de Reconocimiento de Patrones en general.

En este trabajo de tesis de doctorado se propone una nueva estrategia Bayesiana de combinación de clasificadores que permite detectar ambigüedades y resolverlas, lo que constituye la novedad y principal contribución de la tesis. Se propone, a su vez, un sistema completo de reconocimiento de patrones en dos niveles, con una arquitectura modular y paralelizable, que utiliza distintas características extraídas de los patrones de entrada según el problema a resolver junto con la estrategia Bayesiana ya mencionada que decide la respuesta del sistema.

Como elementos componentes del reconocedor, en una primera capa o nivel, se utilizan clasificadores relativamente sencillos y bien posicionados para el problema a tratar. Los elementos pertenecientes a la segunda capa se utilizan para estimar cuán confiable es la respuesta de cada clasificador individual frente a un patrón de entrada, permitiendo decidir cuándo un patrón debe ser considerado bien definido o ambiguo, y en este último caso con qué clases podría confundirse. Adicionalmente, se proponen y aplican estrategias de selección de clasificadores en la etapa de construcción del reconocedor.

El sistema reconocedor de patrones presentado fue aplicado al problema del reconocimiento de dígitos manuscritos off-line, como forma de testear su desempeño. En función de esto, se proponen descriptores basados en características de multirresolución a través del uso de la Transformada Wavelet CDF

9/7 y de Análisis de Componentes Principales, que permiten disminuir considerablemente el tamaño del patrón de entrada y aumentar la calidad de la representación.

La experimentación se realizó sobre las bases de datos CENPARMI y MNIST, ampliamente referenciadas para este problema. Se obtuvieron altos porcentajes en el reconocimiento que alcanzaron un 97,40 y 99,32 % para las bases CENPARMI y MNIST respectivamente. Dichos valores son comparables a los resultados publicados considerados representativos.

Palabras Clave: Reconocimiento de Patrones, clasificación de dígitos manuscritos, patrones ambiguos, redes neuronales, máquinas de soporte vectorial, estadística bayesiana.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción y Objetivos

El Reconocimiento de Patrones es el estudio de cómo las máquinas pueden observar el ambiente o entorno, aprender a distinguir patrones de interés a partir de la experiencia, y tomar decisiones razonables con respecto a las categorías a las que pertenecen dichos patrones. Aunque las investigaciones en este campo llevan más de cincuenta años, el diseño de un reconocedor de patrones artificial de propósito general permanece como una meta lejana. El mejor reconocedor de patrones conocido hasta ahora es el ser humano, no sabiéndose a ciencia cierta cuál es el proceso mediante el cual los humanos reconocemos patrones. Podríamos decir que un patrón es una entidad a la que se le puede dar un nombre [1] [2], como por ejemplo, una imagen de huella digital, una palabra manuscrita, un rostro, una señal representando voz hablada, entre otros muchísimos ejemplos posibles. Dado un patrón, la tarea de su reconocimiento o clasificación puede ser resuelta, en principio, de dos maneras: de forma supervisada, en la cual el patrón será identificado como miembro de una clase predefinida, o de forma no supervisada, en la cual el patrón será asignado a una clase desconocida previamente y aprendida en base a la similitud entre patrones.

El área de Reconocimiento de Patrones representa un desafío en sí misma dentro de la Inteligencia Artificial. Más allá de esto, el interés por la misma se ha visto incrementado sobre todo por la demanda de aplicaciones computacionales relacionadas con diversas áreas como por ejemplo, Minería de Datos (*DataMining*), clasificación de documentos, pronósticos financieros, organización y recuperación en bases de datos multimedia, Biometría (identificación de personas), herramientas para la toma de decisiones, por ejemplo, para diagnósticos médicos, entre muchas otras.

El Reconocimiento Óptico de Caracteres (OCR) es uno de los tópicos más antiguos dentro del Reconocimiento de Patrones. En sus comienzos, la problemática OCR parecía de fácil tratamiento y resolución. Sin embargo, el problema del reconocimiento de caracteres, aunque ha sido estudiado durante varios años obteniendose una alta precisión en las respuestas, está lejos de considerarse resuelto: la precisión en el reconocimiento asociada tanto a caracteres impresos en una imagen degradada o a caracteres

manuscritos, es aún insuficiente. Los métodos existentes basados en aprendizaje no funcionan bien sobre grandes conjuntos de datos categorizados, o sobre conjuntos que crecen, es decir, van incorporando nuevas muestras; los errores en el reconocimiento aún existen, por mencionar algunos de los problemas persistentes. En particular, en los últimos años se han publicado diferentes algoritmos orientados a la clasificación y extracción de características para aplicaciones OCR, técnicas que luego fueron ampliamente utilizadas debido a su buen rendimiento. Se podría afirmar que OCR es una de las áreas de investigación más importante y activa en el ámbito del Reconocimiento de Patrones.

El reconocimiento de caracteres puede dividirse en dos grandes áreas: escritura en línea u *on-line* y escritura fuera de línea u *off-line*. Los reconocedores *on-line* reciben datos a reconocer a medida que los usuarios escriben. En general, tienen que procesar y reconocer la escritura manuscrita en tiempo real o casi en tiempo real. Los reconocedores *off-line* actúan una vez que los datos han sido recolectados y donde, por ejemplo, las imagenes de la escritura manuscrita se ingresan al sistema como mapas de bits. En estos sistemas la velocidad de procesamiento no depende de la velocidad de escritura del usuario, sino de las mismas especificaciones del sistema. Los reconocedores *on-line* están ampliamente difundidos en computadoras manuales, también denominadas *PDA*, donde no hay lugar para un teclado, y en cierta manera son más fáciles de construir que los *off-line*, ya que en la escritura en tiempo real se cuenta con información importante como el orden de los trazos.

Los reconocedores de escritura manuscrita *off-line* juegan un rol importante a partir del hecho de la existencia de grandes cantidades de papel escrito a procesar en nuestra sociedad. Un ejemplo típico de aplicación es el reconocimiento de códigos postales; inclusive bases de datos ampliamente utilizadas para testear sistemas de reconocimiento en el ámbito científico han surgido a partir de esta problemática, que se extiende al reconocimiento de direcciones postales manuscritas.

El Reconocimiento de Dígitos Manuscritos es un tópico destacado dentro del Reconocimiento de Caracteres, que ha recibido desde sus comienzos una importante atención. Con la aparición de las nuevas tecnologías en el campo de las Ciencias de la Computación, y con la explosión de las aplicaciones relacionadas con la manipulación de datos vía Internet, la conversión automática de información escrita y hablada a formularios que puedan ser leídos por una computadora se ha convertido en una tarea de importancia creciente. Por esta razón, las aplicaciones relacionadas con el reconocimiento de números manuscritos, como el procesamiento automático de cheques bancarios, la clasificación de correo en base a la lectura de códigos postales, la lectura automática de formularios y documentos en general con escritura manuscrita, dispositivos de lectura para ciegos, reconocimiento de escritura en computadoras manuales *PDA*, han mantenido a este tema en el centro de las investigaciones.

Además de esto, el Reconocimiento de Dígitos Manuscritos constituye un problema modelo que incluye desafíos comunes con otros tópicos del área de Reconocimiento de Patrones. Por esta razón, es tomado como referencia para la aplicación y testeo de nuevas teorías y algoritmos del área de Reconocimiento de Patrones en general.

La clasificación de números manuscritos es una tarea difícil debido a la gran variedad de estilos de escritura, trazos y orientaciones de los dígitos.

Pensemos que muchas veces ni aún la misma persona que ha escrito un texto entiende su propia escritura. Una de las principales dificultades en el reconocimiento radica en que la varianza intraclase es grande debido a las diferentes formas asociadas a un mismo patrón generadas por el estilo particular de escritura de cada individuo. Esto da lugar a la aparición de patrones que fácilmente pueden ser confundidos con muestras de una clase distinta a la que realmente pertenecen. A pesar de que numerosas investigaciones y modelos se presentan continuamente para este problema, ninguno logra obtener un nivel de respuesta similar al humano.

El objetivo general de este trabajo de tesis de doctorado es presentar un sistema reconocedor de patrones de alto rendimiento que permite detectar patrones ambiguos y en cierta manera ”explicar” las respuestas dadas, valiéndose de la información manejada por el sistema durante las distintas etapas que lleva la clasificación. La arquitectura del sistema es modular y paralelizable, y utiliza distintas características extraídas de los patrones de entrada según el problema a resolver junto con una estrategia Bayesiana para decidir la respuesta del sistema.

La idea subyacente del modelo consiste en utilizar elementos componentes del sistema relativamente sencillos y bien posicionados para el problema a tratar, residiendo el fuerte de la propuesta en el diseño del reconocedor y en la estrategia de definición de respuestas. Esto posibilitó obtener un sistema con características que lo distinguen de los clasificadores reportados en la literatura. El sistema reconocedor de patrones propuesto fue aplicado al problema de reconocimiento de dígitos manuscritos *off-line*, como forma de testear su desempeño.

Como objetivos secundarios que ayudaron a la concreción del objetivo general, mencionamos:

(i) Estudio de la problemática del Reconocimiento de Patrones y en particular del Reconocimiento de Dígitos Manuscritos, orientado a presentar un modelo que realice un aporte frente a las técnicas actuales.

(ii) Estudio del estado del arte en cuanto a descriptores utilizados para este problema específico, en la etapa de preprocesamiento.

(iii) Definición de descriptores adecuados para la clasificación, teniendo en cuenta la importancia de lograr un compromiso entre la capacidad de representar características que permitan discernir entre los elementos de las distintas clases y la dimensión del descriptor. Propuesta de nuevos descriptores para el reconocimiento de dígitos manuscritos que permita mejorar los resultados de la clasificación.

(iv) Estudio de la problemática de la clasificación y propuesta de una estrategia eficiente. El tratamiento de patrones ambiguos y la explicación de las respuestas fue uno de los temas donde se puso énfasis, por su originalidad y las múltiples ventajas que presenta, mejorando la calidad de la respuesta.

(v) Implementación del modelo propuesto y comparación con trabajos publicados. Hemos utilizado herramientas de software de acceso libre ampliamente difundidas, así como también programas desarrollados especialmente. Asimismo, utilizamos las bases de datos de dígitos manuscritos CENPARMI [3] y MNIST [4] [5], las cuales constituyen un estándar dentro de esta temática para testear distintos sistemas de clasificación.

En cuanto a la organización de este trabajo de tesis, la idea es presentar en primer lugar una visión general del modelo, para luego ir desarrollando en detalle cada componente del mismo junto con sus fundamentos, y, finalmente, describir el modelo completo, presentar los resultados obtenidos comparándolos con resultados de la literatura, y exponer las conclusiones del presente desarrollo.

De esta manera, en el Capítulo 2 se trata el tema del estado del arte en cuanto al Reconocimiento de Patrones orientado al Reconocimiento de Caracteres y, en particular, al Reconocimiento de Dígitos Manuscritos. Además, se describe la estructura clásica de un sistema reconocedor de patrones así como también el diseño del reconocedor propuesto.

El Capítulo 3 presenta métodos de clasificación basados en técnicas de aprendizaje ampliamente difundidos en la tarea de Reconocimiento de Patrones, debido a sus beneficios en cuanto a rendimiento en la clasificación. En particular, se tratan técnicas basadas en Redes Neuronales Artificiales y en modelos estadísticos. Asimismo se presenta la experimentación asociada con cada uno de los modelos estudiados aplicados al reconocimiento de dígitos manuscritos.

En los Capítulos 4 y 5 se presentan métodos de extracción de características representativos y bien posicionados para el reconocimiento de dígitos manuscritos, y se proponen nuevos descriptores con

alto rendimiento para el problema en cuestión. Los métodos tratados se basan en la extracción de características direccionales, en la jerarquización de la resolución con que se representa la entrada, y en el análisis multirresolución. Los descriptores propuestos surgen de la aplicación de las distintas técnicas y su combinación. Se presenta, a su vez, la experimentación asociada.

El Capítulo 6 describe la estrategia alternativa de diseño que consiste en combinar múltiples clasificadores individuales para conformar un reconocedor de patrones. Se presentan las estrategias clásicas de combinación así como también la estrategia Bayesiana *EBA* propuesta en este trabajo, junto con la experimentación asociada al problema del reconocimiento de dígitos manuscritos.

En el Capítulo 7 se presenta la experimentación asociada al sistema completo, incluyendo la propuesta de estrategias de selección de clasificadores. Se seleccionan los sistemas finales y se comparan los resultados con los de la literatura.

Finalmente, el Capítulo 8 expone las conclusiones de todo el trabajo y los desafíos a futuro.

Capítulo 2

Reconocimiento de Patrones

El objetivo de este capítulo es presentar el estado del arte en cuanto a Reconocimiento de Patrones orientado al Reconocimiento de Caracteres y, en particular, al Reconocimiento de Dígitos Manuscritos, describiendo a su vez las problemáticas existentes y distintos enfoques estudiados. Además se describe la estructura clásica de un sistema reconocedor de patrones analizando diferentes aspectos y problemáticas sin resolver. En base a todo lo expuesto se presenta el diseño general del sistema reconocedor propuesto.

2.1. Estado del Arte

La disponibilidad de técnicas de aprendizaje ha sido un factor fundamental para el desarrollo y éxito de ciertas aplicaciones de Reconocimientos de Patrones tales como el reconocimiento del discurso y el reconocimiento de la escritura manuscrita [6].

Los métodos utilizados en los comienzos de las investigaciones en OCR fueron *Template Matching* y Análisis Estructural, también conocido como *Feature Matching*. Los templates o prototipos eran diseñados artificialmente y también seleccionados o promediados de las pocas muestras de datos. A medida que el número de muestras se incrementaba, estos métodos se volvían insuficientes para representar la variabilidad de las formas de las muestras, y por lo tanto no permitían construir un clasificador de alta precisión. Para poder aprovechar las ventajas de usar grandes conjuntos de datos, la comunidad científica dedicada al Reconocimiento de Caracteres orientó sus investigaciones a los métodos de clasificación basados en aprendizaje, especialmente las Redes Neuronales Artificiales (RNA) a finales de los 80 y durante la década de 1990. Debido a la estrecha relación entre las RNA y los métodos estadísticos de reconocimiento de patrones, estos últimos también fueron considerados ya que permitían mejorar los resultados. Actualmente, métodos de aprendizaje más nuevos, en particular las Máquinas de Soporte

Vectorial y en forma más general los métodos denominados *Kernel*, así como también los métodos basados en la combinación de múltiples clasificadores, son activamente estudiados y aplicados en el área de Reconocimiento de Patrones [7].

El aprendizaje mediante un conjunto de ejemplos es una característica deseable en la mayoría de los sistemas del área. Los cuatro enfoques más importantes para el Reconocimiento de Patrones son: Template Matching, Clasificación Estadística, Correspondencia Sintáctica o Estructural y Redes Neuronales Artificiales [2] [8]. Estos enfoques no son necesariamente independientes y muchas veces, el mismo método puede ser visto con diferentes interpretaciones desde distintos enfoques. Enmarcados en estos cuatro modelos, se han presentado numerosas técnicas para el problema del reconocimiento de caracteres *off-line*.

El enfoque denominado Template Matching es uno de los más sencillos y está orientado a determinar el grado de similitud entre dos entidades del mismo tipo, que pueden ser puntos, curvas u otras formas. Para esto se debe disponer de un prototipo asociado con el patrón a reconocer (en general de dos dimensiones) y que se aprende a partir de los datos de entrenamiento. Las técnicas de Template Matching pueden agruparse en tres categorías: Correspondencia Directa, Prototipos Deformables y Correspondencia Elástica, y Correspondencia Relajada [9].

En el enfoque Estadístico, cada patrón está representado en términos de d características o mediciones, constituyendo un punto en el espacio de d dimensiones. El objetivo es seleccionar aquellas características que permitan que los patrones que pertenezcan a distintas categorías ocupen regiones compactas y disjuntas en el espacio de características d-dimensional, de forma tal de poder separar los elementos de cada clase adecuadamente. Para esto, y en base a un conjunto de patrones de entrenamiento, se establecen límites de decisión en este espacio de características, por ejemplo, en base a las distribuciones de probabilidad de los patrones de cada clase [2]. Varias de las técnicas más utilizadas para el problema de la escritura manuscrita pertenecen a este grupo, como por ejemplo, el vecino más cercano (*k-Nearest-Neighbor*) [10], el clasificador Bayesiano [11], el clasificador discriminante polinomial [12], Modelos Markovianos Ocultos (*Hidden Markov Model* - HMM) [13] [14], Máquinas de Soporte Vectorial (*Support Vector Machines*) [15] [16], entre otras [8].

En las técnicas correspondientes al Reconocimiento de Patrones Sintáctico, se establece una analogía formal entre la estructura de los patrones y la sintaxis del lenguaje. Los patrones se consideran como estructuras u oraciones del lenguaje, mientras que las primitivas o subpatrones elementales constituyen el alfabeto, de forma tal que estas estructuras u oraciones son generadas de acuerdo a una gramática. Así, un conjunto de patrones complejos se puede describir utilizando un pequeño número de primitivas y reglas gramaticales. La gramática asociada a cada clase se infiere del conjunto de patrones de entrenamiento. El enfoque sintáctico presenta algunas dificultades como, por ejemplo, la necesidad de utilizar grandes conjuntos de datos y estar asociado a altos costos computacionales [17] [18].

Por otro lado, las Redes Neuronales Artificiales tienen la característica de poder aprender correspondencias no-lineales complejas entre valores de entrada y salida, entrenarse de forma automática mediante ejemplos y poder aprender a partir de grandes bases de datos, presentando un muy buen rendimiento frente a datos con ruido. Las RNA han sido ampliamente utilizadas en el Reconocimiento de Patrones y en particular para el problema de Dígitos Manuscritos obteniéndose un alto rendimiento. El Perceptrón Multicapa [19] [20] entrenado con el algoritmo de *Backpropagation* [4] es uno de los modelos clásicos de RNA más estudiados y utilizados [21]. Otras arquitecturas exitosas son las Redes Convolucionales [22], Mapas Auto-organizados de Kohonen o *SOM* [23], *Radial Basis Functions* [24], *Time Delay Neural Networks* [25], entre otros [8].

Todos los enfoques mencionados tienen sus ventajas e inconvenientes. Con el objeto de mejorar los resultados en el reconocimiento aprovechando los beneficios de cada técnica, se han desarrollado distintas estrategias de combinación de clasificadores demostrándose experimentalmente que algunas de ellas realmente mejoran el rendimiento del mejor clasificador individual [26] [27] [8]. El uso de un módulo verificador que refina la elección de la clase de salida entre los mejores candidatos, es otra de las estrategias orientadas a incrementar el porcentaje de patrones correctamente clasificados [13] [28].

El Reconocimiento de Dígitos Manuscritos se enmarca dentro de OCR, y como ya hemos mencionado, es un campo de investigación en continuo desarrollo, debido no sólo a las aplicaciones potenciales sino también a que las soluciones encontradas pueden aplicarse a otros problemas de Reconocimiento de Patrones.

Básicamente las investigaciones en este campo se orientan a los siguientes aspectos: métodos de extracción de características, métodos de clasificación, y sistemas reconocedores basados en diferentes estrategias como, por ejemplo, orientadas a la combinación de clasificadores o a la utilización de módulos verificadores [8].

La investigación en métodos de extracción de características ha ganado una importante atención debido a que el hecho de contar con un conjunto de características que permita discriminar entre patrones de distintas clases, tiene fuerte impacto en el resultado final de la clasificación. En general, los métodos de extracción de características para el problema del reconocimiento de dígitos manuscritos pueden agruparse según traten características estadísticas o estructurales. Las primeras se derivan de la distribución estadística de los puntos de la imagen, como momentos e histogramas. Las características estructurales se basan en propiedades geométricas y topológicas del carácter, como trazos y sus direcciones, intersecciones de segmentos y ciclos.

Mencionaremos trabajos que utilizan extractores de características que consideramos representativos y que constituyen una referencia útil y necesaria para este trabajo de tesis.

Los detectores de bordes de Kirsch [29] han sido utilizados con éxito como extractores de características direccionales para caracteres manuscritos en numerosos trabajos [30] [31] [32] [33] [34], ya

que permiten la detección localizada de segmentos de línea, descripción apropiada para el problema. Este extractor de características es considerado uno de los métodos estándar referenciado en la literatura para este problema. Por otro lado, existen numerosas aplicaciones de la Transformada Wavelet [35] [36] como extractor de características aprovechando el hecho de que los detalles de la imagen en diferentes niveles de resolución caracterizan diferentes estructuras físicas del carácter [25]. En el artículo [37] se presenta un conjunto de descriptores de formas apropiados para representar caracteres manuscritos, basado en la utilización de la transformada wavelet discreta 1D aplicada sobre el contorno del caracter normalizado. La clasificación se lleva a cabo utilizando un conjunto de reconocedores implementados con MLPs.

En [38] se utiliza la familia biortogonal de wavelets *Cohen-Daubechies-Feauveau* (CDF) como extractores de características para representar las variaciones locales en números manuscritos. La experimentación se llevó a cabo sobre la base CENPARMI con las bases CDF 2/2, CDF 2/4, CDF 3/3 y CDF 3/7 en su versión bidimensional, obteniendo descriptores con las cuatro subbandas normalizadas del primer nivel de resolución de las transformadas, mientras que la clasificación se resolvió utilizando la red neuronal multicapa con agrupamiento (*Multilayer Cluster Neural Network*), con un aprendizaje basado en *Backpropagation*.

En [39] se aplica una transformada multiwavelet 1D sin decimación sobre el contorno de los dígitos para construir el descriptor y clasifica utilizando un MLP.

En [40] se aplican técnicas de multirresolución a través de la utilización de la transformada wavelet discreta 2D sobre los dígitos hasta tres niveles de resolución, y se clasifica combinando múltiples MLP entrenados con la técnica de selección dinámica de muestras.

En [41] se utiliza características derivadas de la aplicación de la transformada de Gabor y de las wavelets Haar, Daubechies4, CDF 5/3 y CDF 9/7 para la descripción de texturas. La clasificación fue resuelta con SVM y SOM, comparándose los resultados obtenidos.

En [42] se utiliza la Transformada Wavelet Continua Mexican Hat discretizada para extraer una versión más pequeña de cada dígito y el Gradiente Wavelet para conformar un vector complementario con características de orientación, gradiente y curvatura a diferentes escalas.

Por otro lado, numerosos trabajos han explorado la conveniencia de un cambio de base en la representación mediante la aplicación de Análisis de Componentes Principales (PCA) [32] [33] lo que permite, sin pérdida de información, jerarquizar la resolución con que se representa la entrada (en términos de la varianza de las proyecciones sobre las componentes) sin descomponer la señal en características estructurales como lo hacen las wavelets. Este método permite reducir la dimensionalidad de los datos y es un método clásico en cuanto al reconocimiento de dígitos manuscritos [43]. El método Análisis de Componentes Independientes (ICA) se considera apropiado para distribuciones no Gaussianas, mientras que *Kernel PCA* permite definir técnicas de extracción de características no lineales [2]. Algunas técnicas

combinan redes neuronales y PCA para extraer características [44].También se han presentado trabajos que con este mismo propósito utilizan la Transformada Wavelet junto con PCA e ICA [14].

Un enfoque diferente a la extracción de características tradicional (donde en una etapa de preprocesamiento previa a la clasificación se definen las características relevantes apropiadas para el clasificador), lo constituye la utilización de sistemas basados en aprendizaje cuidadosamente diseñados que operan directamente sobre las imágenes, realizando el análisis de la imágen junto con la extracción de características, para luego clasificar todo dentro de un mismo módulo. Un ejemplo muy exitoso en cuanto a porcentajes de patrones correctamente clasificados para el problema del reconocimiento de dígitos manuscritos, es la red neuronal Convolucional LeNet5 desarrollada por LeCun [4]. Sin embargo, este tipo de enfoques llevan a obtener sistemas complejos en su diseño y dedicados a una tarea específica.

En [45] se presenta un extractor de características entrenable para el reconocimiento de dígitos manuscritos, basado en la red neuronal convolucional LeNet5. La clasificación es realizada por SVMs para mejorar el rendimiento de la red convolucional. Además, el conjunto de entrenamiento es modificado agregando nuevas muestras generadas a partir de las existentes a través de transformaciones afines y distorsiones elásticas. Presenta, además, un análisis de los errores.

La precisión de un sistema de reconocimiento depende fuertemente de dos elementos fundamentales: la capacidad de las características extraídas de representar patrones de cada clase de forma tal de poder ser correctamente discriminados, y el poder de generalización del clasificador utilizado.

La utilización de Redes Neuronales Artificiales ha permitido obtener muy buenos resultados en reconocimiento de caracteres manuscritos. Varios de los trabajos publicados utilizan los métodos clásicos de reconocimiento, como las redes *feedforward* multicapa entrenadas con *Backpropagation* (MLP) [38] [33] [34] [46] [43]. Esta arquitectura ha sido reconocida como una herramienta poderosa para la clasificación de patrones, dado su poder discriminativo y su capacidad de aprender y representar conocimiento implícito [37] [47] [39]. Le Cun [4] presenta una revisión de varios métodos aplicados al problema de OCR y los compara en la tarea del reconocimiento de dígitos manuscritos como caso de estudio, extendiendo el análisis al reconocimiento de documentos. El artículo está orientado a métodos de clasificación que utilizan técnicas de aprendizaje basadas en gradiente, incluyendo las redes convolucionales y MLP, aunque la comparación involucra también otras estrategias de aprendizaje.

También es posible obtener resultados competitivos utilizando técnicas de aprendizaje no supervisado como los mapas auto-organizados de Kohonen [48] [23], para el reconocimiento de números manuscritos [31] [32] [33] [41] [49], aun combinados con otras técnicas [50].

La utilización de los métodos denominados *Kernel* entre los que se incluye las Máquinas de Soporte Vectorial (SVM) [15] han permitido obtener un alto rendimiento en los sistemas para el reconocimiento de patrones y en particular para el problema de dígitos manuscritos [33] [41] [22] [45].

En [46] se compara el rendimiento de SVM virtuales, que utilizan distintos métodos para generar vectores soporte, con otros clasificadores y para bases de datos de dígitos manuscritos estándar.

En [16] se utilizan SVMs para reconocer *strings* de números manuscritos, y se comparan los resultados sobre las bases de dígitos convencionales y también con sistemas basados en MLP, con y sin módulo verificador.

En [34] se propone un clasificador eficiente de tres niveles con estructura de cascada, para el reconocimiento de dígitos manuscritos, que combina técnicas de RNA (MLP) y SVM. Para la extracción de características utiliza Máscaras de Kirsch.

En [51] se propone un método de clasificación híbrido basado en prototipos y en SVM, estas últimas utilizadas para examinar las respuestas dadas por el método basado en prototipos. Este modelo aplicado al reconocimiento de dígitos manuscritos demostró reducir considerablemente los tiempos de entrenamiento y testeo para grandes conjuntos de datos obteniendo un rendimiento comparable a métodos que sólo utilizan SVM.

En cuanto a la combinación de múltiples clasificadores, diversas propuestas han sido publicadas. En [26] se presenta un estudio de diferentes métodos de combinación de clasificadores, y los aplica al reconocimiento de números manuscritos. Experimentalmente muestra que los resultados de la combinación mejoran significativamente el rendimiento de los clasificadores individuales.

Los trabajos [52] [53] [54] [55] presentan un estudio del comportamiento y rendimiento de la estrategia de Voto por Mayoría y variantes, aplicado al problema del Reconocimiento de Patrones. La experimentación es realizada con bases de datos de dígitos manuscritos ampliamente difundidas.

En [25] se presenta un caso de estudio sobre la combinación de clasificadores para el reconocimiento de dígitos manuscritos comparando estructuras de combinación paralela y secuencial. Los clasificadores utilizados se basan principalmente en RBF, MLP y TDNN, mientras que para el preprocesamiento de los patrones aplica la transformada wavelet Haar 2D y extrae características estructurales.

Suen [56] examina los principales métodos de combinación de clasificadores, desarrollados para diferentes niveles de salidas asociadas a los clasificadores individuales: nivel abstracto, nivel de listas por ranking y nivel de mediciones. Analiza diferentes resultados y aplicaciones, entre las que se encuentran el reconocimiento de dígitos manuscritos.

En [57] se presenta un estudio de métodos que generan automáticamente clasificadores a partir de uno base, como Bagging y AdaBoost, y se utilizan distintas estrategias de votación para la combinación de los clasificadores. Se aplica al problema del reconocimiento de texto manuscrito en cursiva. La clasificación está basada en HMM.

En [43] se presenta un método de fusión de clasificadores basado en prototipos de decisión aplicado al reconocimiento de dígitos manuscritos y se lo compara con métodos de combinación clásicos. En el

proceso de extracción de características se aplica PCA para reducir dimensionalidad, y se utilizan MLP como clasificadores individuales.

Creemos importante mencionar que el aprendizaje incremental utilizado para adaptar los clasificadores utilizados a nuevas clases y a nuevas muestras de datos, no ha sido muy considerado en el reconocimiento de caracteres. Por otro lado, el entrenamiento de modelos utilizando datos sin rotular es otro tópico que ha sido intensamente estudiado en el área de Aprendizaje Automático, denominado aprendizaje semisupervisado. Estas técnicas están orientadas a aplicaciones donde la obtención de muestras sin rotular es fácil, y en cambio, el obtener un conjunto de datos rotulados se hace difícil y costoso, como en el caso del reconocimiento de texto [58].

Los siguientes trabajos constituyen una referencia interesante como resumen del estado del arte en el reconocimiento de patrones, y, en particular, para el reconocimiento de dígitos manuscritos.

En [59] y [60] se presenta un estudio de las características de los métodos de clasificación que han sido aplicados con éxito al problema de OCR y se muestran los problemas que aún quedan por resolver. La experimentación es realizada para el problema de dígitos manuscritos utilizando bases de datos entre las más representativas en la literatura.

En [32] y [33] se realiza un estudio comparativo de resultados para el reconocimiento de números manuscritos incluyendo las bases MNIST y CENPARMI, y utilizando técnicas bien conocidas de extracción de características y clasificación para el problema en cuestión. Como extractores de características se utilizan las Máscaras de Kirsch y Sobel, entre otros métodos, y se reduce dimensionalidad usando PCA. Se utilizan clasificadores basados en MLP, SVM, LVQ y RBF entre otros.

En [2] se presenta un estudio y revisión de la problemática del Reconocimiento de Patrones con sus distintos enfoques y métodos más exitosos y representativos, y en particular se trata el enfoque estadístico.

En [46] se realiza un análisis de errores cometidos por varios clasificadores para el problema de dígitos manuscritos sobre las bases de datos más conocidas en la literatura, lo que incluye a las bases CENPARMI y MNIST, y utilizando métodos de clasificación clásicos, como SVM y MLP.

En [22] se presenta la comparación y combinación de técnicas consideradas clásicas y de buen rendimiento para el problema de reconocimiento de dígitos manuscritos para la base MNIST, alcanzando uno de los porcentajes más altos para esta base de datos. Se presenta además un análisis estadístico de los resultados obtenidos.

2.2. Diseño y Estructura de un Sistema Reconocedor de Patrones

En esta Sección trataremos aspectos a tener en cuenta en el diseño de un sistema clasificador de patrones, presentando el estado del arte sobre el tema.

2.2.1. Estructura clásica de un Sistema Reconocedor

El enfoque tradicional para la construcción de un sistema reconocedor de patrones consiste en dividir al sistema en dos módulos principales: un módulo encargado de la extracción de características y el otro dedicado a la clasificación, como muestra la Figura 2.1.

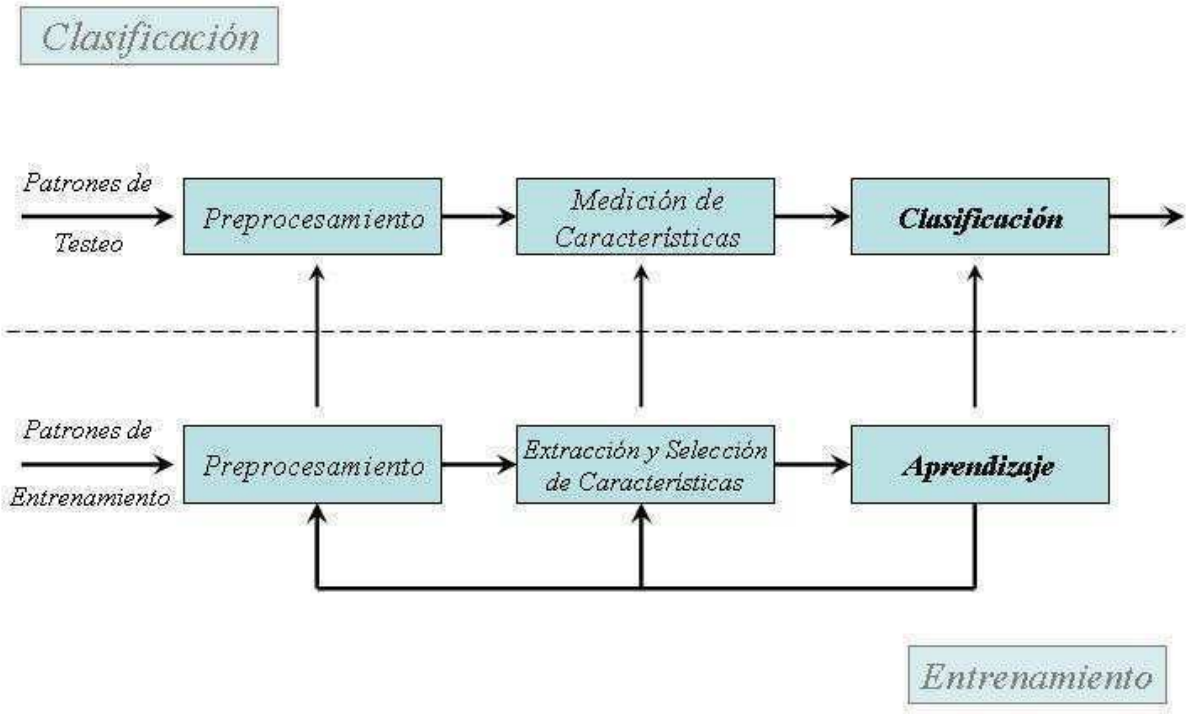


Figura 2.1: Modelo de un sistema reconocedor de patrones siguiendo el enfoque tradicional [2]. Se indican las etapas dentro de las fases de entrenamiento y uso del sistema como clasificador.

El módulo encargado de realizar la extracción de características transforma los patrones de entrada de forma tal que puedan ser representados por vectores de baja dimensionalidad. Esta transformación puede responder a diversas causas, como por ejemplo: obtener ciertas propiedades deseables como invarianza frente a transformaciones y distorsiones de los patrones de entrada, obtener características que permitan discriminar entre los patrones de distintas clases aún los más difíciles, permitir un buen rendimiento del clasificador. Este módulo extractor de características posee conocimiento previo acerca del problema, específico a la tarea a resolver. Es una etapa importante en el diseño del sistema reconocedor, de fuerte impacto en los resultados finales y sobre la que se desarrollan numerosas investigaciones. Muchas veces, el término *preprocesamiento* hace referencia no sólo a la etapa previa a la extracción de características, sino que incluye también a este proceso.

El módulo dedicado a la clasificación es usualmente entrenable y de propósito más general.

Utilizando el reconocimiento de escritura manuscrita como un caso de estudio, algunos autores muestran que la extracción de características también puede ser implementada por sistemas basados en aprendizaje cuidadosamente diseñados, que operan directamente sobre las imágenes, realizando el análisis de la imágen junto con la extracción de características, para luego clasificar todo dentro de un mismo módulo. Un ejemplo muy exitoso en cuanto a porcentajes de patrones correctamente clasificados para el problema del reconocimiento de dígitos manuscritos, es la red neuronal Convolucional LeNet5 desarrollada por LeCun [4]. Sin embargo, este tipo de enfoques llevan a obtener sistemas complejos en su arquitectura y dedicados a una tarea específica. Por otro lado, los sistemas donde cada tarea se trata en módulos independientes permiten mayor flexibilidad y una aplicación más general del sistema clasificador, y es el enfoque más utilizado. El presente trabajo de tesis está orientado a este último tipo de clasificadores con una etapa de preprocesamiento y extracción de características bien diferenciada de la etapa de clasificación y definición de la salida del sistema.

Como hemos mencionado en la Sección anterior, el proceso de definición y extracción de características constituye un área de investigación muy activa dentro del Reconocimiento de Patrones y del Aprendizaje Automático. La clasificación de caracteres ha sido mayormente resuelta utilizando un número limitado de características seleccionadas artificialmente, en general, por el diseñador del sistema en base a conocimiento previo del problema a resolver. Si se incrementa la cantidad de características a considerar, esto puede complicar el diseño del clasificador deteriorando su capacidad de generalización. Actualmente se están considerando técnicas para seleccionar automáticamente de un gran conjunto de características candidatas, un buen conjunto de características que permita obtener una mejor clasificación que el conjunto seleccionado manualmente.

Jain [2] remarca la diferencia entre *selección* y *extracción de características*, ya que los dos son aspectos importantes a tener en cuenta. La *selección* de características se refiere a algoritmos que seleccionan los subconjuntos considerados mejores del conjunto de características de entrada. La *extracción*

de características se refiere a métodos que crean nuevas características en base a transformaciones o combinaciones aplicadas sobre el conjunto de datos o características originales. Notar que, frecuentemente, la extracción de características precede a la selección de las mismas ya que se seleccionan cuáles serán las características más convenientes para la clasificación, de las extraídas. Algunos de los métodos más conocidos para la selección de características son: Búsqueda Exhaustiva que evalúa todos los subconjuntos posibles; *Branch and Bound*; seleccionar las mejores características individuales (lo cual no asegura obtener un subconjunto óptimo); *Sequential Forward Selection*; *Sequential Backward Selection*. Los métodos de selección envolventes utilizan el error de clasificación de un subconjunto de características para medir su efectividad. Otros, denominados filtros, realizan la selección de características en la etapa de preprocesamiento [61].

Con respecto a los métodos de clasificación utilizados en el reconocimiento de caracteres, éstos pueden agruparse en métodos basados en vectores de características y métodos estructurales (ver Sección 2.1). Los métodos basados en vectores de características prevalecen sobre los segundos, especialmente para el reconocimiento de caracteres *off-line*, debido a su implementación más sencilla y su menor complejidad computacional. Como métodos en este grupo podemos mencionar los métodos estadísticos, redes neuronales, máquinas de soporte vectorial y combinación de múltiples clasificadores [7], de los cuales nos ocuparemos en este trabajo.

El objetivo de mejorar los porcentajes de reconocimiento de los clasificadores individuales, combinando múltiples clasificadores, ha sido perseguido durante mucho tiempo. En [55] se presenta un estudio de estos métodos aplicados al reconocimiento de caracteres, mencionando distintas formas de organizar los clasificadores. Por ejemplo, una combinación paralela u horizontal se adopta frecuentemente para obtener una precisión alta, mientras que una organización secuencial o en cascada se utiliza principalmente para acelerar los tiempos en una clasificación que involucra un gran número de clases.

De acuerdo al nivel de información de la salida de los clasificadores, las estrategias de fusión para los métodos de combinación en paralelo pueden categorizarse en: nivel abstracto, nivel de rangos y nivel de mediciones, como veremos en el Capítulo 6. En [56] se presentan resultados sobre la combinación de múltiples clasificadores combinados en diferentes niveles.

El rendimiento en la clasificación no sólo depende de la estrategia de combinación, sino que también depende de la complementariedad o diversidad de los clasificadores seleccionados. Esta complementariedad puede obtenerse, por ejemplo, variando las muestras de entrenamiento, las características extraídas, la estructura de los clasificadores, los métodos de aprendizaje, entre otras opciones. En los últimos años se desarrollaron métodos para generar múltiples clasificadores a través de la exploración de la muestras del conjunto de entrenamiento en función de una característica específica, los cuales están recibiendo una creciente atención, como es el caso de las técnicas de Bagging [62] y Boosting [63]. Para

el problema del reconocimiento de caracteres, la combinación de clasificadores basada en la utilización de distintas técnicas de preprocesamiento y extracción de características ha demostrado ser efectiva.

La comparación de clasificadores no es tarea sencilla dado que muchos modelos, en particular las redes neuronales artificiales, son flexibles en cuestiones de implementación y su rendimiento puede verse afectado por factores humanos. En el área del reconocimiento de caracteres la comparación de métodos de clasificación se hace más difícil debido a la existencia de varias etapas como el preprocesamiento y extracción de características y luego la clasificación. Para la comparación es recomendable, entonces, además de utilizar los mismos conjuntos de entrenamiento y testeo, utilizar técnicas estándar en las etapas involucradas en el sistema de clasificación, excepto en el proceso que se quiere comparar.

En la escritura manuscrita sin restricciones y en particular en los dígitos manuscritos, podemos encontrar grandes variaciones en las formas de representar los datos. Un número manuscrito puede estar representado por un trazo tan pobre que apenas se lo puede asociar con su versión impresa, o inclusive con otro número. Esta es una de las posibles razones de error en la clasificación en los humanos (lo llamaríamos un dígito inclasificable). Sin embargo, en el conjunto de datos mal clasificados por un sistema automático solemos encontrar patrones que para un humano están bien definidos y serían perfectamente clasificables, mientras que otros, en contrario, se muestran ambiguos y/o distorsionados. Es deseable que el subconjunto de datos clasificables para un ser humano sea correctamente reconocido por un sistema automático.

Uno de los desafíos que plantea el reconocimiento de patrones en general, y en particular el reconocimiento de dígitos manuscritos, es el tratamiento de patrones ambiguos y *outliers*. Los patrones ambiguos pertenecen a alguna de las categorías o clases definidas para el problema, pero pueden confundirse entre varias clases. Los patrones denominados outliers no pertenecen a ninguna clase de las definidas (por ejemplo, un patrón resultado de una mala segmentación de la imagen). Varios de los sistemas de clasificación presentados en la literatura proponen la opción de rechazo para patrones que no están claramente asociados con una clase de las predefinidas, incluyendo muchas veces a los ambiguos además de los outliers. Algunos trabajos tratan específicamente el tema de los outliers [64] [65] [66].

Suen [46] presenta un estudio basado en el análisis de errores cometidos por múltiples clasificadores para el problema del reconocimiento de dígitos manuscritos. En este estudio se presentan tres categorías posibles de causas de error en la clasificación de los datos:

Categoría 1: asociada a imágenes de dígitos que se confunden fácilmente con patrones de otras clases debido a la similitud de sus características primitivas y estructura. Como es sabido, algunos pares de números son más fáciles de confundir que otros, como por ejemplo ocurre con patrones de las clases 4 y 9, 0 y 6, 3 y 5. Las imágenes en esta categoría podrían pertenecer a alguno de estos pares.

Categoría 2: asociada a imágenes de dígitos que son difíciles de identificar aún para un humano. Esta dificultad en la identificación puede deberse a ruido en la imagen, deformaciones en el dígito o mala segmentación del carácter, entre otras posibles causas. Por ejemplo, la distorsión y degradación de la imagen podría tener origen en el grosor del trazo debido al instrumento de escritura utilizado, al tamaño elegido para normalizar las imágenes, el tipo de escritura no convencional de algunas personas.

Categoría 3: asociada a imágenes de dígitos que son fácilmente clasificables por humanos sin ninguna ambigüedad. Las imágenes en esta categoría son claras y sería esperable que fuesen bien clasificadas.

Según este estudio, alrededor de un cuarto de los datos mal clasificados pertenecen a la Categoría 1. De alguna manera es esperable que estas imágenes sean mal clasificadas dado que sus formas son ambiguas, es decir, tienen estructuras similares a las de otras clases, pero con características locales diferentes. Si se analizaran estas características locales algunos de estos números podrían ser correctamente clasificados. Sin embargo, debido a la ambigüedad de los patrones y siendo algunos aún confusos para el ser humano, existe un subconjunto de imágenes donde es difícil de determinar la clase a la que pertenecen, es decir, es difícil decidir si el rótulo correcto es el provisto en la base de datos o el obtenido por el clasificador. De hecho, *frente a algunas imágenes las personas pueden tener diferentes opiniones acerca de qué dígito están observando*.

Con respecto a la Categoría 2, el número de muestras mal clasificadas correspondiente a esta categoría es pequeño. En general, distintos autores recomiendan rechazar este tipo de patrones en la clasificación, debido a que se supone no podrían asociarse con ninguna clase [46].

El mayor porcentaje de imagenes mal clasificadas está asociado con la Categoría 3, donde los humanos no tendrían inconveniente en identificar correctamente a dichos números. En algunos casos, estos errores se explican en la falta de muestras suficientes para determinada estructura de número asociada a determinada clase. Por ejemplo, sabemos que hay dos formas bien diferentes de escribir un cuatro, si éstas no están presentes en el conjunto de entrenamiento será difícil que luego el clasificador pueda asociar ambas con la clase del cuatro. Por otro lado, algunas de estas muestas son claras y tienen un formato estándar de número, con lo cual son candidatas a ser bien clasificadas.

De acuerdo a este análisis, y en términos generales, en distintos trabajos se han planteado estrategias para disminuir el error. Una de ellas es la utilización de un módulo verificador, cuya tarea consiste en evaluar más precisamente los resultados producidos por el clasificador y de esta forma compensar sus debilidades. Según Takahashi y Griffin [67] existen tres tipos de verificadores: verificación absoluta (pertenece el patrón a la clase del 1?), verificación entre dos clases (es un 3 o un 5?), y verificación en grupos (es un 0, 6 ó 9?). Por ejemplo, la verificación entre dos clases serviría para atacar los errores

asociados con la Categoría 1, sabiendo de antemano cuáles son las clases que suelen confundirse y construyendo dichas combinaciones de verificadores.

Otra estrategia posible y efectiva para disminuir el error es la combinación de múltiples clasificadores, que ya hemos mencionado. Según Suen [46], si se realiza un análisis en cuanto a ventajas, limitaciones y complementariedad de los clasificadores individuales utilizados y se aplica una regla de combinación adecuada, podría lograrse una alta tasa de disminución del error.

2.3. Sistema Reconocedor propuesto

En esta Sección presentaremos el diseño general del Sistema CLasificador de patrones con tratamiento de AMbigüedad, *SCLAM*.

El Sistema Reconocedor propuesto responde a una estructura tradicional, donde se diferencian claramente dos etapas: preprocesamiento y extracción de características, y la clasificación que incluye la definición de la salida del sistema.

La etapa de preprocesamiento y extracción de características comprende el tratamiento previo de los datos de entrada con técnicas de normalización del tamaño de la imagen y eventual binarización si correspondiera, para luego extraer características consideradas relevantes para el problema a tratar, en este caso de aplicación al tratamiento de dígitos manuscritos. La elección de características la realiza el experto, diseñador del sistema, en base a su conocimiento. Para esta etapa hemos trabajado con características tradicionales y bien posicionadas para el problema, como aquellas que detectan las direcciones de los trazos de línea, las que jerarquizan la resolución con que se representa la entrada y permiten reducir la dimensionalidad, y aquellas que surgen de un análisis de multirresolución de la imagen. A su vez, proponemos nuevos descriptores basados en la aplicación de las técnicas descriptas y su combinación, asociados a un alto rendimiento en cuanto a la discriminación entre patrones de distintas clases. El tema de extracción de características será desarrollado en los Capítulos 4 y 5.

En la etapa de clasificación, el sistema reconocedor comprende dos niveles.

El primer nivel está formado por un conjunto de elementos clasificadores independientes y paralelos, cada uno dedicado a una característica diferente. El segundo nivel consiste en un módulo analizador encargado de definir y, de alguna manera, explicar la salida del sistema. Este módulo utiliza los siguientes elementos: la tabla de confiabilidad y dos parámetros ajustables durante la etapa de puesta a punto del reconocedor. Cada elemento clasificador del primer nivel produce una respuesta frente a un patrón de entrada dado, como si fuera un juez que decide a qué clase pertenece el patrón en base al análisis de la característica a la que está dedicado. La conexión entre los dos niveles del sistema se realiza a través de la nueva representación del patrón de entrada, formada por el voto de los ”jueces” o clasificadores

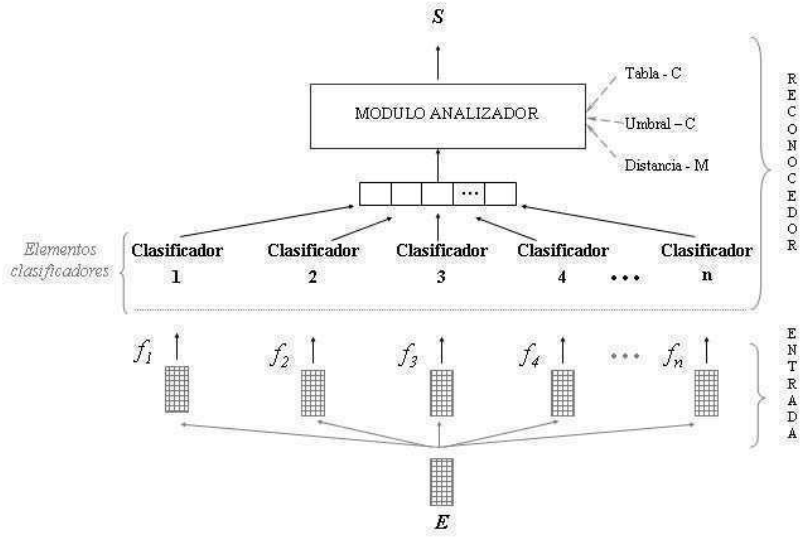


Figura 2.2: Arquitectura del reconocedor propuesto. f_i indica la característica utilizada, luego asociada con un clasificador individual; n indica la cantidad total de clasificadores. En el módulo analizador: Tabla de Confiabilidad (Tabla-C) y parámetros umbral de confiabilidad (Umbral-C) y distancia mínima (Distancia-M).

individuales. La tabla de confiabilidad juega un papel fundamental en la estrategia para definir la salida del sistema. La misma expresa cuán confiable es cada voto emitido por cada elemento clasificador, permitiendo realizar un voto calificado.

El sistema es capaz de explicar sus respuestas, indicando a qué clase se parece más el patrón de entrada en función de cada característica analizada. Como parte de la explicación, si un patrón es considerado ambiguo por el sistema, podremos saber con qué otras clases podría confundirse, es decir, con qué otras clases tiene más características en común. Además, el diseño en su conjunto permite aumentar la precisión en la clasificación como lo muestra la experimentación presentada en el Capítulo 7.

La Figura 2.2 muestra la estructura general del sistema reconocedor para más de cuatro clasificadores, a modo de ejemplo.

A continuación, analizaremos las características que surgen de este diseño. La estructura modular y paralelizable del primer nivel del reconocedor presenta varias ventajas, entre las que podemos mencionar: la posibilidad de agregar y/o eliminar elementos clasificadores fácilmente, sin tener que reentrenar todo el sistema y sólo modificando la tabla de confiabilidad y los parámetros ajustables en el módulo analizador. A su vez, la tabla de confiablidad permite observar la conveniencia de mantener o no algún elemento

clasificador, según su rendimiento asociado con cada clase. Por ejemplo, si un clasificador es muy bueno para una clase en particular y para otras no tanto, por la forma en que está definida la estrategia de combinación en el módulo analizador, podría ser beneficioso incorporar este clasificador. A su vez, el paralelismo e independencia de los clasificadores permite realizar un aprendizaje en simultáneo, disminuyendo los tiempos de entrenamiento. El Capítulo 7 desarrolla el tema de los clasificadores individuales utilizados.

Por otro lado, el hecho de contar con varios elementos clasificadores que después se combinan para obtener una respuesta, permite aumentar en el sistema la tolerancia a errores, es decir, si algunos clasificadores se equivocan en la respuesta, esto podría no afectar la respuesta final dada la participación de otros clasificadores más precisos para el patrón de entrada analizado.

Una de las características del modelo que nos interesa reamarcar es el tratamiento de patrones ambiguos.El sistema detecta si un patrón es ambiguo o no, y en el caso de serlo determina con qué otra(s) clase(s) se podría confundir. Además, la estrategia implementada permitiría definir criterios para distinguir entre patrones ambiguos y outliers: por ejemplo, si todos los clasificadores están en desacuerdo con la clase a la que pertenece un patrón, éste podría ser tratado como outlier. Otra posibilidad es considerar la opción de rechazo (significando ”es outlier”) como una clase más en cada clasificador individual, y asociado a un criterio en el módulo analizador para tratar estos casos. Este tema se retomará en el Capítulo 7, con ejemplos y análisis de la salida de los sistemas reconocedores.

En general, los sistemas presentados en la literatura rechazan los patrones que no están claramente asociados a una clase y a los otros los asocian con una clase en particular. Pero los humanos no respondemos únicamente de esta manera, y así lo reporta [46] en su análisis de errores.

La salida generada por el sistema propuesto en este trabajo intenta acercarse un poco más a las respuestas que un humano pudiera dar. Es decir, informar cuándo un patrón no es del todo claro, y a qué clases podría pertenecer, diciendo ”este patrón es confuso, podría ser un 3 o un 5”. Además, el sistema puede explicar cómo llegó a la respuesta final en base a todos los elementos que intervienen en las decisiones tomadas. Por ejemplo, el sistema podría decir que un patrón determinado pertenece a cierta clase porque los clasificadores asociados con ciertas características lo asociaron a dicha clase y esos clasificadores no suelen equivocarse en casos similares.

La estrategia Bayesiana propuesta se describe en el Capítulo 6, mientras que los detalles de implementación del sistema en su conjunto se presentan en el Capítulo 7 junto con los resultados intermedios y finales.

Para cerrar este Capítulo, diremos que la idea general que siguió el desarrollo del trabajo fue obtener un sistema de alto rendimiento con una calidad de respuesta diferente de los clasificadores tradicionales, basado en técnicas relativamente sencillas y bien posicionadas para el problema de aplicación, residiendo el fuerte de la propuesta en el diseño del reconocedor y en la estrategia de definición de respuestas.

Los resultados intermedios obtenidos nos han permitido también generar otras propuestas originales para el problema particular tratado, como la definición de descriptores para el reconocimiento de dígitos manuscritos que han permitido mejorar los resultados de los clasificadores individuales, reduciendo el costo computacional e impactando positivamente en todo el sistema reconocedor.

Capítulo 3

Métodos de Clasificación

El objetivo de este capítulo es presentar métodos de clasificación basados en técnicas de aprendizaje, ampliamente difundidos en la tarea de Reconocimiento de Patrones dados sus beneficios en cuanto a rendimiento en la clasificación. Se tratarán técnicas basadas en Redes Neuronales Artificiales y en modelos estadísticos. Asimismo se presenta la experimentación asociada con cada uno de los modelos estudiados aplicados al reconocimiento de dígitos manuscritos.

3.1. Introducción

Los métodos de clasificación basados en técnicas de aprendizaje, y en especial las Redes Neuronales Artificiales (RNA), han llamado la atención entre los investigadores del área de Reconocimiento de Patrones, principalmente durante las décadas de 1980 y 1990. El motivo fue poder aprovechar las ventajas de utilizar grandes conjuntos de datos para el entrenamiento de los sistemas para mejorar así la precisión en el reconocimiento. Junto con las RNA, los métodos estadísticos de clasificación también fueron ampliamente considerados en este período [11]. En la actualidad, en el área de Reconocimiento de Patrones se está estudiando y aplicando activamente métodos de aprendizaje como las Máquinas de Soporte Vectorial entre los métodos *Kernel*, y los sistemas que combinan múltiples clasificadores. En cuanto al reconocimiento de caracteres, los métodos basados en aprendizaje permitieron mejorar significativamente el rendimiento de los sistemas de clasificación. Sin embargo, y aunque ha habido importantes avances, el problema del reconocimiento de caracteres no ha sido resuelto: la precisión de los sistemas actuales frente a imágenes distorsionadas o frente a escritura manuscrita es insuficiente y dista de la precisión desarrollada por un humano.

Los métodos de clasificación para el reconocimiento de caracteres pueden categorizarse en *métodos basados en vectores de características* y *métodos estructurales*. Los primeros están más difundidos

especialmente para el reconocimiento de caracteres *off-line*, debido a su simple implementación y bajo costo computacional [7]. En los métodos estructurales los caracteres son representados como uniones de estructuras primitivas. Estos métodos pueden categorizarse en gramaticales y gráficos [8]. En este capítulo abordaremos métodos de clasificación basados en vectores de caracterísiticas relevantes en el reconocimiento de caracteres en general y en el reconocimiento de dígitos manuscritos en particular. Los mismos estarán basados en la utilización de RNA y SVM. Las técnicas para combinar múltiples clasificadores serán tratadas en el Capítulo 6.

3.2. Redes Neuronales Artificiales

Las Redes Neuronales Artificiales o Redes Neuronales, constituyen un paradigma computacional alternativo al tradicional basado en la programación de secuencia de instrucciones. El enfoque conexionista está inspirado en conocimientos provenientes de la Neurociencia pero sus métodos provienen de diversas disciplinas como la Física Estadística, Psicología, Teoría del Conocimiento, Teoría de Sistemas, entre otras. Sus aplicaciones potenciales se orientan principalmente al campo de las Ciencias de la Computación y de la Ingeniería, entre las que se pueden mencionar: la extracción de características desde un conjunto de datos complejos (ej.: imágenes, discurso); aplicaciones orientadas al reconocimiento óptico de caracteres (OCR) y procesamiento de imágenes; implementación directa y paralela de algoritmos de búsqueda y correspondencia; problemas que deben manejar datos contradictorios, difusos, probabilísticos e incompletos. Cabe destacar que las redes neuronales están particularmente bien situadas con respecto a las aplicaciones para reconocimiento de patrones [68].

3.2.1. Perceptrón Multicapa

Las redes neuronales del tipo Perceptrón Multicapa (MLP) han sido utilizadas en las últimas décadas en los sistemas OCR. Estas redes pueden comportarse como clasificadores y también como extractores de características. La Figura 3.1 muestra un ejemplo de arquitectura, donde cada nodo o neurona en una capa está totalmente conectado con los nodos de las capas previa y siguiente. Las capas de neuronas que conforman este tipo de redes son: la capa de unidades de entrada, una o varias capas ocultas, y la capa de unidades de salida. Durante la etapa de entrenamiento, los pesos asociados al conexionado son modificados de forma tal que la red neuronal ”aprenda”. Uno de los algoritmos de aprendizaje más utilizado denominado de *Retropropagación hacia atrás* o *Backpropagation* (BP) [4] [24], utiliza la técnica de descenso por gradiente para encontrar el mínimo de la función de costo que mide el error del sistema como una función diferenciable de los pesos. El enfoque estocástico tiene la ventaja de permitir una amplia

exploración de la superficie de costo: los patrones de entrenamiento son presentados a la red en orden aleatorio, modificándose los pesos luego de presentar cada patrón.

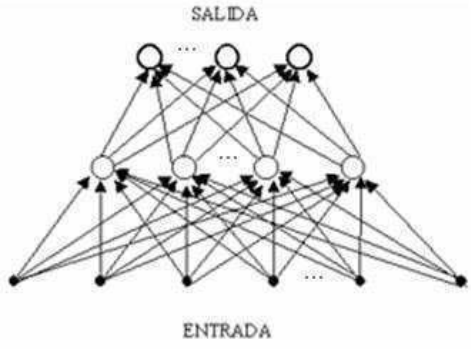


Figura 3.1: Ejemplo de una arquitectura de un perceptrón multicapa con una capa oculta

Este procedimiento tiende a decrementar la función de costo en cada iteración (para valores de parámetros adecuados), hasta que la misma se adapte al gradiente local. Otra de las ventajas del enfoque estocástico es que en diferentes ejecuciones del algoritmo, el resultado es diferente, permitiendo sortear los mínimos locales asociados a un error alto con finalización del proceso de aprendizaje.

Para cada iteración o tiempo t, se define:

- w_{ij} sinapsis que conecta a la neurona i de la capa m con la neurona j de la capa $m-1$
- $v_i = \sum_{j \in J} w_{ij} y_j$ entrada neta a la neurona i donde J incluye a todas las neuronas de la capa anterior a la de la neurona i
- $y_i = \varphi(v_i)$ salida de la neurona i, donde φ es la función de activación. Si y_i está en la capa de entrada de la red entonces toma los valores del patrón ingresado.
- ζ_i salida esperada en la neurona i (cada patrón está rotulado ya que el algoritmo se enmarca en el paradigma de *Aprendizaje Supervisado*)
- O_i salida real para la neurona i
- $E(t) = \frac{1}{2} \sum_{i \in C} (\zeta_i - O_i)^2$ donde C incluye a todas las neuronas de la capa de salida de la red, es el error cuadrático medio en la iteración t

En base a esto la la función de costo queda determinada por:

$$E(w) = \frac{1}{N} \sum_{\mu \in P, i \in C} (\zeta_i^\mu - O_i^\mu)^2 \tag{3.1}$$

donde C incluye a todas las neuronas de la capa de salida de la red y P incluye a todos los patrones que conforman el conjunto de entrenamiento. Notar que O_i es de la forma $\sum_{j \in J} w_{ij} y_j$ luego, es correcto que E esté definida en función del vector de pesos w. La misma es una función diferenciable y tiene un minimo absoluto. Para minimizarla se utiliza la técnica del descenso por gradiente, la cual consiste en alterar los pesos en la dirección que produce el máximo descenso en la superficie de error. La dirección de cambio se obtiene mediante el gradiente (derivadas parciales de la función de costo con respecto a los pesos) ya que el mismo especifica la dirección que produce el máximo incremento, por lo que el mayor descenso es el negativo de esa dirección.

Para el enfoque estocástico, los pesos se actualizan de acuerdo a la siguiente regla [69]:

- $w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t)$
- $\Delta w_{ij}(t) = -\eta \frac{\partial E(t)}{\partial w_{ij}(t)} + \alpha \Delta w_{ij}(t-1)$

El parámetro η es el coeficiente de *velocidad de aprendizaje*, que debe ser elegido en forma adecuada ya que la convergencia de la red es muy suceptible a su valor. Una velocidad de aprendizaje muy alta puede producir oscilaciones en la convergencia de la red, mientras que una velocidad muy pequeña puede llevar a una convergencia muy lenta. Un método para incrementar la velocidad de aprendizaje y evitar el riesgo de inestabilidad (oscilación en la convergencia) es la inclusión del parámetro α en la regla de aprendizaje llamado *coeficiente de momentum o inercia* el cual hace que el cambio realizado en los pesos sea en la dirección de descenso promedio, acelerando la convergencia. Otra ventaja de utilizar este coeficiente de momentum es la posibilidad de saltear mínimos locales durante el proceso de entrenamiento ya que el método del descenso por gradiente garantiza alcanzar un mínimo de la función de error pero no necesariamente el absoluto.

Existen múltiples variantes para el diseño y entrenamiento de una red neuronal usando el algoritmo BP, orientadas a evitar o disminuir el efecto de ciertos problemas inherentes al método en cuestión. Por ejemplo, durante el entrenamiento el parámetro η puede permanecer constante o no, dando origen a dos clases de *backpropagation*. La segunda alternativa es una mejora con respecto a la primera y la idea principal para llevarla a cabo es mantener un valor de η tan alto como sea posible mientras que el aprendizaje sea estable, es decir, no oscilante. Un ejemplo de cómo implementar esto es el siguiente: si el error se va decrementando entonces se aumenta el valor de η en un factor η_{inc} y se actualizan los pesos, pero si el error se incrementa entonces el valor de η se decrementa en un factor de η_{dec} y se descarta la actualización de pesos en esa iteración [69]. Este último enfoque generalmente se denomina *BP con velocidad de aprendizaje adaptativa*.

Otra cuestión delicada es la inicialización de los pesos antes del entrenamiento, ya que una elección inadecuada de valores puede causar que el aprendizaje finalice en un mínimo local. Algunos autores [30] proponen el uso de algoritmos genéticos para definir la inicialización.

También se han propuesto otras arquitecturas de red como alternativa a la estándar, consistente en al menos una capa oculta y donde cada unidad de una capa se conecta con todas las unidades de la capa siguiente, con el objeto de mantener la cantidad de pesos acotada y evitar el problema del sobreentrenamiento. Un ejemplo de arquitectura alternativa entrenada con BP es la red *Multilayer Cluster Neural Network* utilizada en varios trabajos [30]. La red Convolucional (*Convolutional Neural Network*) es otro ejemplo de arquitectura alternativa que utiliza conexiones locales y pesos compartidos y que ha dado excelentes resultados en el reconocimiento de caracteres manuscritos y de dígitos manuscritos en particular. Su estructura permite trabajar directamente sobre la imagen, ya que las neuronas ocultas con conexiones locales actúan como extractores de características entrenables. Pero este tipo de redes tienen una estructura mucho más compleja y dedicada al problema específico a tratar [7] comparadas con las redes dedicadas únicamente a la tarea de clasificar.

Una de las variantes más importantes del algoritmo BP la da el método que establece cómo modificar los pesos. El enfoque clásico denominado *Batch* calcula la dirección de descenso (el gradiente) luego de presentar todos los patrones a la red en cada época del entrenamiento, es decir, modifica los pesos luego de calcular el gradiente exacto. La variante es el enfoque *Estocástico* ya mencionado, donde el gradiente se aproxima en base a una muestra (o a un pequeño número de muestras) del conjunto de entrenamiento, para luego modificar los pesos usando este gradiente aproximado.

La versión Batch tiene asociada la problemática en la cual el encuentro de un mínimo local finalice el aprendizaje con error alto. Por otro lado, permite aplicar otras técnicas para encontrar la dirección de descenso, como Gradientes Conjugados o algoritmos como Levenberg-Marquardt o Gauss-Newton. Sin embargo, estos métodos no son apropiados para entrenar con grandes conjuntos de datos y sobre arquitecturas con un número considerable de conexiones [4] principalmente por la velocidad de convergencia. La versión Estocástica ha demostrado ser superior en cuanto a velocidad de convergencia y rendimiento en la clasificación en el tratamiento de grandes conjuntos de datos que presentan redundancia.

De esta manera, la red neuronal Perceptrón Multicapa entrenada con el algoritmo de Backpropagation, junto con sus variantes, se ha transformado en una herramienta estándar a la hora de clasificar y comparar resultados, debido principalmente a su destacada capacidad de discriminar patrones pertenecientes a distribuciones complejas [70].

3.2.2. Mapas Autoorganizados de Kohonen (SOM)

Los Mapas Autoorganizados de Kohonen o SOM (*Self-Organizing Maps*) son redes neuronales no supervisadas que se caracterizan principalmente por generar mapeos o correspondencias desde un espacio de señales de alta dimensionalidad a estructuras topológicas de baja dimensionalidad. Estas correspondencias son capaces de preservar las relaciones de vecindario de los datos de entrada y tienen la característica de representar regiones de alta densidad en las señales de entrada como grandes zonas de la estructura topológica resultante. Por otro lado, el conjunto de datos de entrada es particionado en subconjuntos o *clusters* que agrupan patrones con características similares. Estas características son de utilidad en aplicaciones de diversas áreas desde el reconocimiento del discurso y compresión de datos hasta el control de un robot.

Desarrollado por Teuvo Kohonen en el año 1982, el SOM está basado en un algoritmo de aprendizaje competitivo y no supervisado. Competitivo ya que utiliza una función de interacción lateral entre unidades, y no supervisado, ya que tiene la capacidad de descubrir la similitud entre patrones de entrada (formación de clusters), sin necesidad de que estos patrones estén rotulados [48].

Un mapa autoorganizado consiste en un conjunto de i unidades o neuronas que habitualmente conforman una grilla de dos dimensiones, y donde cada unidad está asociada a un vector de pesos $w_i = [v_{i_1}, \ldots, v_{i_m}] \in \Re^m$.

Cada elemento $x \in \Re^m$, perteneciente al espacio de entrada de alta dimensionalidad, es presentado a la red. Es decir, por cada elemento de entrada se calcula la activación de cada neurona para luego seleccionar la unidad ganadora o BMU (*Best Matching Unit*), que es el nodo asociado con el vector de pesos w_* con menor distancia a la entrada presentada (ver fórmula 3.2).

$$\text{dist}(x, w_*) = \mathbf{min}_{i=1\ldots n}(\text{dist}(x, w_i)) \tag{3.2}$$

El siguiente paso del algoritmo de entrenamiento consiste en disminuir la diferencia entre el vector de entada y el vector de pesos asociado a la unidad ganadora w_*, modificando éste último en una fracción de la distancia indicada por la velocidad de aprendizaje α, parámetro que decrece en el tiempo. Los vectores de pesos asociados a las unidades vecinas a la ganadora, también se modifican, según una función de vecindario $h_{i,*}$ que también decrece en el tiempo.

La regla de aprendizaje del algoritmo se define como:

$$w_i(t+1) \quad := \quad w_i(t) \; + \; \alpha(t)\, h_{i,*}(t)\, (x(t) \; - \; w_i(t)) \tag{3.3}$$

donde t es la iteración actual en el proceso de aprendizaje, α representa la velocidad de aprendizaje, $h_{i,*}$ es la función de vecindario, x el vector de entrada, y w_i es el vector de pesos asociado con la neurona i.

Este algoritmo de entrenamiento permite obtener un ordenamiento topológico de los vectores de entrada presentados a la red durante el proceso de aprendizaje. De esta forma, datos de entrada similares se corresponderán con regiones vecinas en el mapa de salida. Este mapa final preserva la similitud de los datos en el espacio de características, agrupando vectores con características similares en neuronas vecinas.

La versión Batch es una variante del algoritmo SOM propuesta por Teuvo Kohonen en 1990 [23]. En la misma, la modificación de los pesos del mapa se realiza una vez presentados todos los patrones a la red, en vez de realizar la adaptación por cada patrón ingresado como ocurre en la versión tradicional. La versión Batch constituye un enfoque determinístico que elimina la utilización del parámetro de velocidad de aprendizaje. En algunos casos, esta variante mejora la eficiencia del algoritmo, sobre todo en cuanto a tiempos y complejidad.

Más allá de sus propiedades destacables, uno de los inconvenientes del SOM reside en que la búsqueda de la BMU domina el tiempo de cómputo del algoritmo, haciéndolo computacionalmente costoso para entradas de alta dimensionalidad y para mapas grandes. Varios modelos fueron presentados para sortear estas dificultades, como las variantes denominadas PicSOM [71] y *Tree Structured SOM* (TS-SOM) [72] [73], entre otras [74]. Algunas de ellas se basan en la utilización de una estructura jerárquica de SOMs obteniendose mejoras sobre los tiempos de búsqueda; sin embargo, la utilización de un único mapa permite obtener muy buenos rendimientos en cuanto al ordenamiento del mismo sobre todo en los límites de las distintas regiones. Es por eso que se han desarrollado modelos que proponen un procesamiento paralelo sobre un único mapa, orientado a la búsqueda de la unidad ganadora lo cual reduce notablemente los tiempos de procesamiento sin disminuir la calidad del mapa final [75] [76] [77].

Otro enfoque del SOM aplicado a reconocimiento de patrones lo constituye la variante supervisada denominada Learning Vector Quantization (LVQ) [78], en la cual el algoritmo aprende los prototipos para cada clase y esto permite clasificar los datos de entrada. La aplicación del SOM y luego el ajuste de los límites de las regiones usando LVQ puede resultar beneficioso [24] [79], aunque en la practica puede ser difícil llegar a un buen ajuste y definir los parámetros en forma adecuada, lo que además puede requerir una inversión en tiempo y recursos elevada.

3.3. Máquinas de Soporte Vectorial

3.3.1. Introducción

Las Máquinas de Soporte Vectorial o *Support Vector Machines* (SVM) fueron propuestas por Vapnik como un sistema de aprendizaje automático basado en la teoría de aprendizaje estadístico [8]. Este método ha captado la atención de los investigadores en las áreas de Aprendizaje Automático y Reconocimiento de Patrones debido principalmente a su exitoso rendimiento en diferentes áreas como el reconocimiento de rostros, categorización textual, predicciones, recuperación de imágenes y reconocimiento de escritura manuscrita. Una de las características destacables es su excelente capacidad de generalización aún en espacios de alta dimensionalidad y utilizando conjuntos de entrenamiento pequeños [16].

Diseñadas originalmente como clasificadores binarios, la idea base reside en construir un hiperplano de separación entre las clases maximizando la distancia (el margen) entre las mismas. Las SVM utilizan una función núcleo para representar el producto interno entre dos patrones, en un espacio de características no lineal y expandido (posiblemente de dimensionalidad infinita). Las etapas de entrenamiento y clasificación se realizan a través de esta función núcleo, sin necesidad de operar en el espacio no lineal.

Para resolver problemas de clasificación con más de dos clases (multiclase), se han planteado estrategias para combinar las SVM binarias. Entre las más difundidas podemos mencionar uno-contra-todos y uno-contra-uno. La estrategia uno-contra-uno ha demostrado funcionar mejor cuando se utilizan funciones kernel lineales mientras que para kernels no lineales la estrategia uno-contra-todos funciona eficientemente [7]. En los últimos años se han publicado varios trabajos que utilizan SVM para el reconocimiento de caracteres y en especial para el reconocimiento de dígitos manuscritos. Los resultados muestran que las SVM permiten obtener altos porcentajes de reconocimiento en comparación con clasificadores estadísticos o basados en redes neuronales [33], pero por otro lado, el costo computacional y de almacenamiento que necesita la utilización de un gran número de vectores soporte hace que este método no siempre sea la mejor elección. Algunos enfoques utilizan un clasificador estadístico o neuronal para seleccionar dos clases candidatas para luego llegar a una decisión final a través de una SVM [80].

3.3.2. Hiperplano de separación óptimo

La idea original de las Máquinas de Soporte Vectorial consiste en usar un hiperplano de separación lineal para clasificar los patrones de entrenamiento en dos clases. Dados los vectores de entrenamiento x_i, $i = 1, ..., l$ de longitud n, y el vector y definido en 3.4, esta técnica busca un plano de separación con el mayor margen entre las dos clases, dicho margen medido sobre la linea perpendicular al hiperplano.

$$y_i = \begin{cases} 1 & \text{si} \quad x_i \text{ está en la clase 1, o} \\ -1 & \text{si} \quad x_i \text{ está en la clase 2} \end{cases} \tag{3.4}$$

La Figura 3.2 muestra un caso de patrones (a) linealmente separables y (b) no linealmente sepa-

(a) (b)

Figura 3.2: SVM (a) Datos linealmente separables con hiperplano; (b) Datos que no son linealmente separables [81]

rables. En el caso (a) puede observarse un ejemplo de dos clases que pueden separarse por la línea punteada $w^T x + b = 0$. La idea es que la distancia de los patrones de entrenamiento de cada una de las dos clases a la línea de separación sea máxima. Es decir, se desea encontrar una linea con parámetros w y b tal que la distancia $w^T x + b = \pm 1$ sea maximizada. Como la distancia entre $w^T x + b = 1$ y $w^T x + b = -1$ es $\frac{2}{\|w\|}$ y maximizar $\frac{2}{\|w\|}$ es equivalente a minimizar $w^T \frac{w}{2}$, entonces se plantea el siguiente problema:

$$min_{w,b} \ \frac{1}{2} w^T w \tag{3.5}$$

sujeto a

$$y_i((w^T x_i) + b) \geq 1, i = 1, ..., l. \tag{3.6}$$

La restricción 3.6 especifica que los datos pertenecientes a la clase 1 deben estar del lado derecho de $w^T x + b = 0$ mientras que los datos pertenecientes a la otra clase deberán estar del lado izquierdo. Notar que la razón de maximizar la distancia entre $w^T x + b = 1$ y $w^T x + b = -1$ se basa en la Minimización del Riesgo Estructural planteado por Vapnik [81].

La formulación Lagrangiana del problema 3.7 tiene ventajas importantes: permite representar las restricciones sobre los multiplicadores de Lagrange, y además los datos de entrenamiento sólo participan

de operaciones de producto interno entre vectores, lo cual permite generalizar el procedimiento para el caso no lineal [82].

$$L_P = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{l} \alpha_i[(y_i(x_i^T w + b) - 1] \tag{3.7}$$

donde α_i son los multiplicadores de Lagrange. La solución se encuentra minimizando L_P con respecto a w y b, requiriendo simultaneamente que las derivadas de L_P con respecto a todos los α_i se anulen, siendo $\alpha_i \geq 0 \; \forall i$.

En forma equivalente, el problema puede resolverse maximizando el Lagrangiano dual con respecto a α_i:

$$\begin{aligned} max L_D = \quad & \textstyle\sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j}^{l} \alpha_i \alpha_j y_i y_j (x_i^T x_j) \\ \text{sujeto a} \quad & \alpha_i \geq 0, i = 1, \ldots, l \text{y} \\ & \textstyle\sum_{i=1}^{l} \alpha_i y_i = 0 \end{aligned} \tag{3.8}$$

Luego, la función discriminante o regla separadora asociada está dada por:

$$f(x) = sign(\sum_{vectoressoporte} y_i \alpha_i (x_i^T x) + b) \tag{3.9}$$

donde los x_i se denominan *vectores soporte* asociados a multiplicadores de Lagrange α_i no nulos. Los vectores soporte son los patrones de entrenamiento ubicados sobre los límites del margen separador [45].

3.3.3. Problema no lineal y datos no separables

Debido a que los problemas de clasificación reales son difíciles de resolver con un clasificador lineal, el modelo fue extendido a superficies de decisión no lineales a través de la utilización de una función núcleo o kernel no lineal como reemplazo del producto interno realizado por el algoritmo lineal [16]. De esta manera, el hiperplano óptimo estará en el espacio de características, el cual surge como una correspondencia no lineal con el espacio de entrada, y en general tiene una dimensionalidad mucho más alta que el espacio de entrada, lo que facilitaría la separación de los datos en clases. Otra de las ventajas de este enfoque es que no es necesario operar en el espacio de características directamente, sino que esto se realiza a través de la función kernel. La fórmula 3.10 expresa la función discriminante derivada de la SVM para un problema de dos clases, para una función núcleo $K(x, x_i)$, siendo x un patrón a clasificar y x_i un patrón de entrenamiento,

$$f(x) = sign(\sum_{vectoressoporte} y_i \alpha_i K(x_i, x) + b) \tag{3.10}$$

donde y_i es el rótulo del patrón de entrenamiento x_i, y los valores de los parámetro α_i y b son determinados maximizando una función cuadrática [45][15]. La Tabla 3.1 muestra las funciones kernel más utilizadas.

Tabla 3.1: Funciones *Kernel* más utilizadas para SVM.

Kernel	Producto interno asociado
Lineal	$K(x, y) = (x \cdot y)$
Gaussiana	$K(x, y) = exp(-\frac{\|x-y\|^2}{2\sigma^2})$
Polinomial	$K(x, y) = (x \cdot y)^p$

Para abordar el problema de datos no separables se utiliza una estrategia que consiste en permitir errores durante el entrenamiento y patrones que quedan dentro del margen de separación. Para esto se introduce un conjunto de variables ξ_i y un parámetro C que permite regular la cantidad de errores aceptados en relación con la maximización del margen de separación:

$$min \frac{1}{2} \|w\|^2 + C(\sum_{i=1}^{l} \xi_i) \tag{3.11}$$

sujeto a

$$y_i((w^T \phi(x_i)) + b) \geq 1 - \xi_i, \xi_i \geq 0, i = 1, ..., l. \tag{3.12}$$

Este nuevo planteo deriva en el mismo problema dual presentado en 3.8, pero con la siguiente condición agregada sobre los multiplicadores de Lagrange:

$$0 \leq \alpha_i \leq C, i = 1, ..., l. \tag{3.13}$$

La definición de un valor para C es una tarea delicada, ya que podría comprometer el rendimiento del clasificador. Por ejemplo, si los errores se penalizaran con un valor muy alto podría producirse un sobreentrenamiento de la SVM [16][45]. Otra alternativa para este problema lo constituye la variante $\nu - SVM$, donde el parámetro C es reemplazado por un parámetro $\nu \in [0, 1]$ que representa una cota inferior sobre el número de vectores soporte y una cota superior sobre el número de ejemplos que están del lado incorrecto del hiperplano [83][84].

La Figura 3.3 muestra un caso de datos (a) linealmente separables y (b) no separables. Los vectores soporte sobre los límites del margen están representados con doble círculo.

(a) (b)

Figura 3.3: SVM - Hiperplanos de separación lineal para el caso (a) separable, (b) no separable. Los vectores soporte se indican con doble círculo [82].

3.3.4. Clasificación Multiclase

Hemos mencionado que las SVM son clasificadores binarios. Para obtener un clasificador de más de dos clases (multiclase), las SVM binarias pueden combinarse a través de diferentes estrategias siendo las más difundidas uno-contra-todos y uno-contra-uno, debido a su buen rendimiento en el problema de la clasificación de dígitos manuscritos. Los clasificadores multiclase basados en SVM son aún un tema de investigación en curso [85] [86] [87]. Las propuestas incluyen también la utilización de una única SVM que clasifica q clases, $q > 2$, en lugar de q SVM binarias que luego se combinan, resolviendo así un único problema de optimización [45].

En el enfoque uno-contra-uno, se construye un clasificador por cada par de clases con el objeto de separar las clases de a dos. Luego, los clasificadores se disponen conformando una estructura de árbol, donde cada nodo es una SVM. De esta forma cada muestra de entrada se compara con cada uno de los pares y el ganador será testeado en el nivel superior del árbol hasta llegar a la raíz. En esta estrategia el número de clasificadores a entrenar es de $\frac{q(q-1)}{2}$, con lo cual para el caso de los dígitos manuscritos donde $q = 10$, deberíamos entrenar 45 SVM.

La estrategia uno-contra-todos construye un clasificador por cada una de las q clases con el objeto de separar cada una del resto. Utiliza un experto F para decidir la respuesta final entre todas las respuestas dadas por las SVM. Sea $h = (h_1, \ldots, h_q)^T$ la salida del sistema de q SVMs. El operador *arg max* selecciona la clase Q para la entrada x tal que maximice h_Q,

$$F = argmax(h) \tag{3.14}$$

Sin embargo, esta estrategia de decisión tiene el problema que no todas las SVM producen una salida en la misma escala. Para solucionar este inconveniente, antes de comparar las respuestas las mismas son normalizadas. Sea $s(h)$ la salida normalizada de un sistema de q SVMs combinadas con la estrategia uno-contra-todos, la regla de decisión se define como [16],

$$F = argmax(s(h)) \tag{3.15}$$

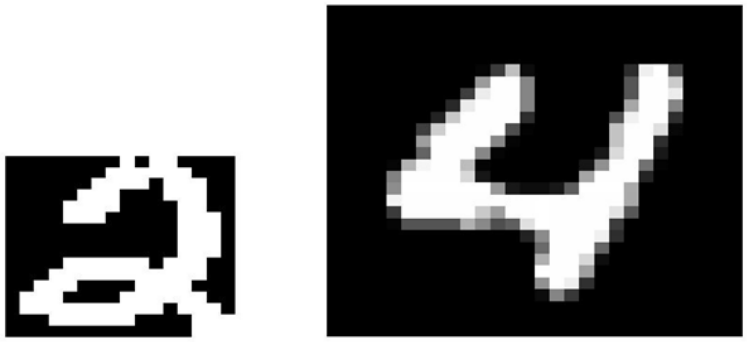
La estrategia uno-contra-uno ha demostrado funcionar mejor cuando se utilizan funciones kernel lineales. Por otro lado, para kernels no lineales la estrategia uno-contra-todos funciona eficientemente [7].

3.4. Aplicación de los métodos de clasificación al problema del reconocimiento de dígitos manuscritos

En esta Sección presentamos la aplicación de los métodos de clasificación desarrollados, sobre las bases de datos CENPARMI y MNIST descriptas en el Apéndice A.

La base CENPARMI está compuesta por imágenes de dígitos binarizadas y de distinto tamaño, que fueron ajustadas a 16x16 píxeles tanto para el conjunto de entrenamiento de 4000 patrones como para el conjunto de testeo de 2000 patrones. En cambio, la base MNIST está compuesta por imágenes en escala de grises de tamaño 28x28, y presenta un conjunto de entrenamiento de 60000 patrones y uno de testeo de 10000. Aunque ambas bases de datos incluyen casos de dígitos con distorsiones y ambigüedades y que son difíciles de clasificar, la base de datos MNIST tiene en su tratamiento la dificultad de el gran volumen de datos a procesar, mientras que la base CENPARMI de menor tamaño y resolución, presenta un subconjunto de dígitos de muy difícil clasificación.

La Figura 3.4 presenta imágenes de ambas bases de datos y compara el tamaño de las mismas. También existen diferencias en la resolución de las imágenes, dado que en la base CENPARMI éstas están binarizadas y en MNIST no. Teniendo en cuenta que a la hora de clasificar dígitos manuscritos básicamente lo que se hace es reconocer la forma de una línea, pareciera natural considerar imágenes con dos tonos, uno para la forma y otro para el fondo, es decir, binarizadas. La binarización estaría descartando todos aquellos valores que no aportan información a la hora de definir el patrón además de resaltar la forma sobre el fondo, con lo cual se esperaría mejorar los resultados. Hemos elegido la media de cada patrón como umbral para binarizar la base MNIST. La Figura 3.5 muestra imágenes en escala de grises y binarizadas.



(a) (b)

Figura 3.4: Comparación del tamaño de las imágenes de las bases (a) CENPARMI y (b) MNIST.

A partir de esto, hemos aplicado los métodos de clasificación Perceptrón Multicapa, Mapas de Kohonen y Máquinas de Soporte Vectorial sobre las bases de datos.

3.4.1. Perceptrón Multicapa

Para la experimentación se ha utilizado el modelo de Perceptrón Multicapa entrenado con Backpropagation, utilizando la técnica de descenso por gradiente con momentum y velocidad de aprendizaje adaptativa. En la definición de la arquitectura de este tipo de redes hemos utilizado una sola capa oculta y la función de activación logística 3.16 en las capas oculta y de salida, que es lo que nos dio mejores resultados.

$$logsig(n) = \frac{1}{(1 + \exp(-n))} \tag{3.16}$$

La Tabla 3.2 presenta los resultados del reconocimiento sobre los conjuntos de testeo de las bases CENPARMI, MNIST, y MNIST binarizada. También se indica la arquitectura de red en cuanto a cantidad de neuronas en la capa de entrada (corresponde a la dimensión del patrón), en la capa oculta y en la capa de salida (diez neuronas correspondientes a la cantidad de clases definida). Por ejemplo, la arquitectura 256 x 160 x 10 corresponde a la red utilizada para el conjunto de entrenamiento CENPARMI con imágenes normalizadas a tamaño 16x16, y con 160 neuronas en la capa oculta.

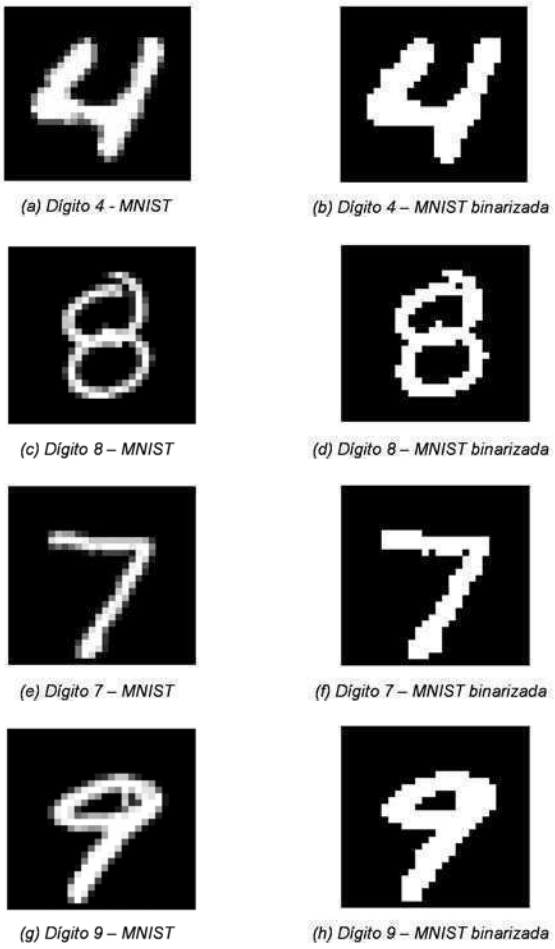


Figura 3.5: Imágenes de MNIST en escala de grises y binarizadas.

La Figura 3.6 muestra, a modo ilustrativo, la evolución del error en función de las épocas para el entrenamiento de la red asociada al conjunto MNIST binarizado.

Observando los resultados de la Tabla 3.2 podemos decir que el porcentaje de patrones correctamente clasificados obtenido para la base CENPARMI es deseable de mejorar, lo que intentaremos hacer utilizando diferentes preprocesamientos junto con otros métodos de clasificación. La utilización del MLP es útil ya que el ajuste de parámetros resulta menos dificultoso comparado con otras técnicas, y sus resultados son lo suficientemente confiables y competitivos a la hora de realizar comparaciones. En cuanto a la base MNIST, ésta presenta la dificultad del gran volumen de datos y la definición de las imágenes lo que impacta fuertemente en la implementación y el rendimiento del clasificador. Para la experimentación hemos extraído un subconjunto de 15000 patrones del conjunto de entrenamiento, seleccionados aleatoriamente de forma tal que cada clase estuviera representada por la misma cantidad de muestras. Este subconjunto de entrenamiento lo utilizaremos a lo largo del trabajo para poder comparar resultados.

Tabla 3.2: Resultados del reconocimiento sobre los conjuntos de testeo de las bases CENPARMI, MNIST, y MNIST binarizada utilizando MLP. Para MNIST y su versión binarizada se utilizó un conjunto de entrenamiento de 15000 patrones.

Base	MLP	% Patrones Testeo reconocidos
CENPARMI	256 x 160 x 10	89.05
MNIST	784 x 100 x 10	76.86
MNIST binarizada	784 x 160 x 10	96.22

El conjunto de testeo fue utilizado en su totalidad. Vemos que el porcentaje de patrones correctamente clasificados mejora notablemente para el caso de la base binarizada, de 76.86 % a 96.22 %. En el caso de la base MNIST con imágenes en escala de gris el clasificador no llega al mínimo de error definido, estancándose la mayoría de las veces en mínimos locales. En cambio, para la base binarizada el clasificador ajusta rápidamente al error definido. Debido a esto, continuaremos con la experimentación teniendo en cuenta la base MNIST binarizada, ya que creemos que la binarización aporta información a la hora de reconocer los dígitos manuscritos.

3.4.2. Mapas Autoorganizados de Kohonen

Los Mapas Autoorganizados de Kohonen tienen la gran ventaja de mostrar gráficamente la distribución de los datos de entrenamiento en un arreglo de salida, generando una correspondencia desde el espacio de entrada de zonas con alta densidad de puntos, a una mayor cantidad de neuronas en el espacio de salida. Para el problema de dígitos manuscritos y para el análisis de la utilización de diferentes preprocesamientos, esto podría resultar útil. A continuación, y a modo de ejemplo, se presenta la experimentación sobre las bases CENPARMI y MNIST.

Para la base CENPARMI, se utilizó un mapa de dimensión 30 x 30 neuronas entrenado durante 1000 épocas. La estructura del mapa fue considerada como un toroide, de forma tal de evitar el efecto borde. Por ejemplo, las neuronas en las esquinas del mapa conforman un sólo grupo o *cluster*. La Figura 3.7 muestra el mapa obtenido luego de las fases de entrenamiento y rotulado de las neuronas, según el criterio de la mayor frecuencia de activación, y eliminando las neuronas que fueron activadas por un número elevado de clases. Puede observarse claramente la formación de agrupamientos según los rótulos de las clases. También puede observarse en los límites de las regiones de cada clase, con qué clases en más propensa a confundirse la clase analizada.

El porcentaje de reconocimiento asociado al mapa de la Figura 3.7 para el conjunto de testeo CENPARMI fue de 88.90 %. El mismo es menor en comparación con el obtenido para MLP. Sin embargo,

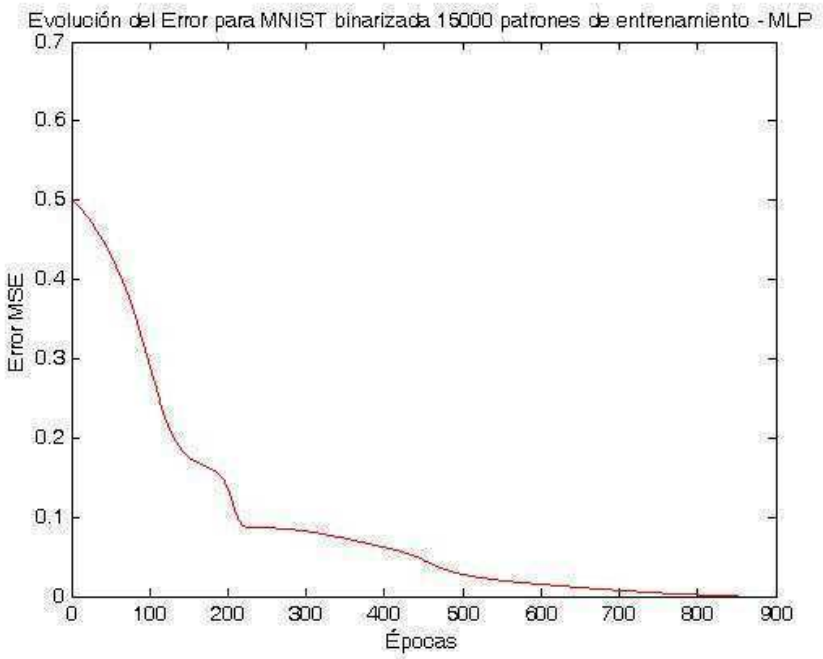


Figura 3.6: Evolución del error en función de las épocas para el aprendizaje de la red asociada al conjunto MNIST binarizado para 15000 patrones de entrenamiento, utilizando MLP con arquitectura 784 x 160 x 10. La red converge, obteniendose el minimo de error definido (0.001) en 852 épocas de entrenamiento.

creemos que los SOM pueden aportar a la hora de construir un sistema clasificador más complejo [49], con lo cual tendremos en cuenta también esta técnica en el Capítulo 6. Una de las dificultades del algoritmo SOM tradicional es el tiempo requerido para el entrenamiento con grandes volúmenes de datos. Y este es el problema que surgió a la hora de entrenar con la base MNIST. Por tal motivo, además de extraer un subconjunto de 15000 patrones de los datos de entrenamiento, se utilizó la base preprocesada con técnicas que serán presentadas en el Capítulo 5, lo que nos permitió reducir la dimensión del descriptor de 784 a 98, y de esta manera disminuír considerablemente el costo computacional. La Figura 3.8 muestra el mapa obtenido luego de 1500 épocas de entrenamiento.

El porcentaje de reconocimiento asociado al mapa de la Figura 3.8, obtenido sobre todo el conjunto de testeo MNIST fue de 94.53 %.

3.4.3. Máquinas de Soporte Vectorial

Las Máquinas de Soporte Vectorial tienen un muy buen rendimiento aplicadas al reconocimiento de dígitos manuscritos, como hemos mencionado en secciones anteriores. A su vez, tienen la dificultad de

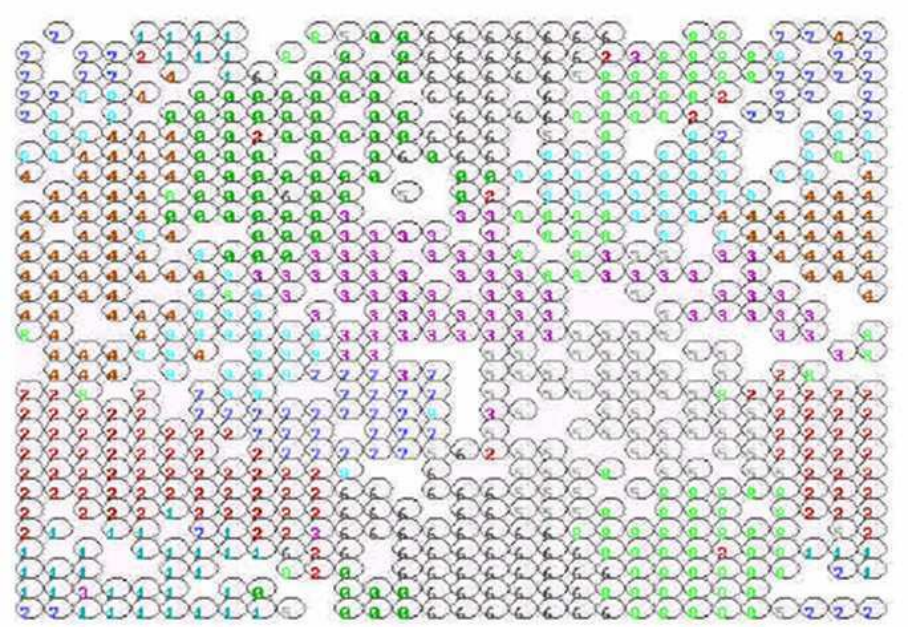


Figura 3.7: Mapa SOM para la base CENPARMI con imágenes ajustadas a 16x16. Se observa la formación de *clusters* en un mapa considerado toroide.

la complejidad del ajuste de parámetros orientado a obtener un buen rendimiento del sistema.

Para la experimentación se ha utilizado el paquete de software libre TORCH [88], herramienta especialmente desarrollada para problemas de alta dimensionalidad y gran cantidad de muestras. En particular se han utilizado SVM gaussianas, dado que reportan los mejores resultados para el problema de reconocimiento de dígitos manuscritos [16] [33].

La Tabla 3.3 muestra los resultados del reconocimiento para las bases CENPARMI y MNIST binarizada, asi como los valores utilizados para el parámetro σ.

Tabla 3.3: Resultados del reconocimiento sobre los conjuntos de testeo de las bases CENPARMI y MNIST binarizada utilizando SVM gaussianas. Para MNIST binarizada se utilizó un conjunto de entrenamiento de 15000 patrones.

Base	SVM σ	% Patrones Testeo reconocidos
CENPARMI	18.0	85.00
MNIST binarizada	14.5	97.33

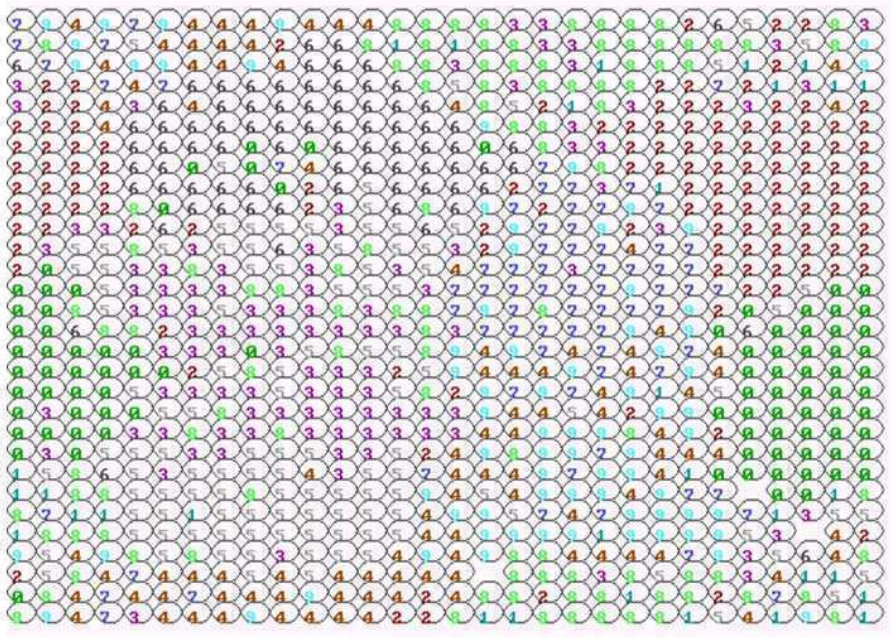


Figura 3.8: Mapa SOM 30 x 30 para 15000 patrones de entrenamiento de la base MNIST con imágenes representadas por el descriptor LL1pca98 (ver Capítulo 5). Se observa la formación de *clusters* en un mapa considerado toroide.

3.4.4. Conclusiones

Los resultados obtenidos en la presente Sección se resumen en la Tabla 3.4. Los mismos nos ayudarán a definir una heurística en función de la cual decidiremos qué clasificadores usar y sobre qué conjunto de datos, sobre todo en el Capítulo 5 donde se proponen nuevos descriptores.

Tabla 3.4: Porcentajes de patrones correctamente clasificados para los conjuntos de testeo de las bases CENPARMI y MNIST binarizada utilizando distintos métodos de clasificación. Para MNIST binarizada se utilizó un conjunto de entrenamiento de 15000 patrones.

Base	MLP	SOM	SVM
CENPARMI	89.05	88.90	85.00
MNIST binarizada	96.22	94.53	97.33

Observamos que la aplicación de MLP y SVM permite obtener los mejores porcentajes de patrones correctamente clasificados sobre ambas bases (esto no desmerece la utilidad del SOM, que como ya mencionamos, tendremos en cuenta en la construcción de un clasificador más complejo, en el Capítulo 6). Teniendo en cuenta que el costo en la etapa de ajuste de parámetros para SVM es alto, elegimos

realizar las pruebas subsiguientes utilizando MLP, y aplicar SVM en los casos donde ya se haya definido el preprocesamiento a utilizar en el marco de la construcción del sistema final.

Capítulo 4

Extracción de Características

El objetivo de este capítulo es presentar métodos de extracción de características representativos y bien posicionados para el reconocimiento de dígitos manuscritos, como base para la búsqueda de un buen descriptor que permita mejorar los resultados en la clasificación, tema que se planterá en el Capítulo siguiente. Los métodos presentados se basan en la extracción de características direccionales, en la jerarquización de la resolución con que se representa la entrada, y en el análisis multirresolución.

4.1. Introducción

El rendimiento de un sistema reconocedor de patrones depende fuertemente de la capacidad discriminante de las características seleccionadas para representar los datos, además de la capacidad de generalización del clasificador utilizado [34].

El proceso de extracción de características no sólo implica la definición de la forma en la cual va a representarse cada patrón sino que también, muchas veces implica reducir la dimensionalidad del descriptor o representante, todo orientado a obtenter un rendimiento competitivo en la clasificación.

Un buen conjunto de características debería cumplir con las siguientes condiciones deseables: tener asociada una varianza intraclase pequeña, con lo cual diferentes muestras de una misma clase estarán cercanas en distancia; generar una separación entre clases (interclase) grande, de forma tal que las muestras de clases diferentes estén lo suficientemente alejadas en el espacio de características, para evitar errores de clasificación. Justamente, una de las dificultades de la escritura manuscrita radica en que la varianza intraclase es grande, debido a las variaciones de forma y trazo asociadas a los diferentes estilos de escritura de las personas. No existe un modelo matemático que pueda describir tales variaciones, por lo tanto, la investigación se orienta a encontrar un conjunto de características que prueben una razonable

insensibilidad a las distorsiones de forma, pero por otro lado mantengan la capacidad de separar muestras de distintas clases [37].

El tipo de características a extraer depende fuertemente del problema a tratar. Para el reconocimiento de dígitos manuscritos diferentes métodos han sido utilizados. Mencionaremos como ejemplo la transformada de Fourier y la Característica Loci, entre otros [43]. Liu [32] realiza un estudio sobre el rendimiento de distintos clasificadores, utilizando diversas características, presentando el estado del arte sobre el tema para el problema del reconocimiento de dígitos manuscritos. En el mismo, las características que utiliza son las direccionales, usando, entre otros, los operadores de Kirsch y de Sobel.

4.2. Extractores de características direccionales: Máscaras de Kirsch

Las máscaras de Kirsch han sido utilizadas como extractores de características direccionales en numerosos trabajos relacionados con el reconocimiento de dígitos manuscritos [30] [31] [32] [33] [34]. Los números, ya sea manuscritos o impresos, están compuestos esencialmente por dibujos de líneas, es decir, estructuras de una dimensión en un espacio bidimensional. Por lo tanto, la detección localizada de segmentos de línea constituye un método de extracción de características considerado como uno de los métodos estándar referenciados en la literatura para este problema.

La elección del detector de bordes de Kirsch sobre otros detectores de bordes diferenciales de primer orden representativos, como el de Frei-Chen, de Prewitt o de Sobel, se basa en que entre todos ellos las máscaras de Kirsch son conocidas por detectar bordes en las cuatro direcciones en forma rápida, adecuada y más precisa que otros, dado que los ocho vecinos son considerados en su totalidad [47]. A su vez, este detector de direcciones de línea es robusto aún en presencia de ruido [34].

Kirsch define el siguiente algoritmo para la extracción de características direccionales, donde para cada posición en la imagen se da información acerca de la presencia de un segmento de línea en cierta dirección [29]:

$$G(i,j) = \text{máx}\left\{1, \max_{k=0}^{7}\left[|5S_k - 3T_k|\right]\right\} \tag{4.1}$$

definiendo

$$S_k = A_k + A_{k+1} + A_{k+2} \tag{4.2}$$

y

$$T_k = A_{k+3} + A_{k+4} + A_{k+5} + A_{k+6} + A_{k+7} \tag{4.3}$$

En la ecuación 4.1, $G(i,j)$ representa el gradiente correspondiente al pixel (i,j), los subíndices de A son evaluados módulo 8, y $A_k(k = 0, 1, \ldots, 7)$ representa cada uno de los ocho vecinos del pixel (i,j) definidos en la Figura 4.1.

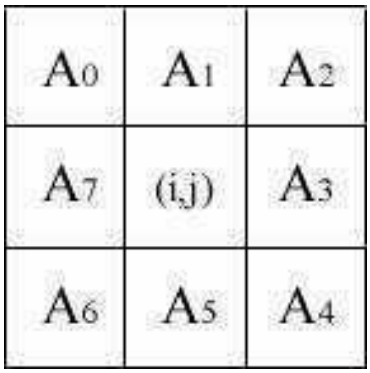


Figura 4.1: Definición de los ocho vecinos del pixel (i,j), denominados $A_k(k = 0, 1, \ldots, 7)$.

Para extraer cada una de las cuatro características direccionales, y en función de las fórmulas 4.1, 4.2 y 4.3, se calcula una nueva representación de la imagen original para las direcciones horizontal (HR), vertical (VT), diagonal derecha (RD), diagonal izquierda (LD), según lo siguiente:

$$G(i,j)_{HR} = \text{máx}(|5S_0 - 3T_0|, |5S_4 - 3T_4|) \tag{4.4}$$

$$G(i,j)_{VT} = \text{máx}(|5S_2 - 3T_2|, |5S_6 - 3T_6|) \tag{4.5}$$

$$G(i,j)_{RD} = \text{máx}(|5S_1 - 3T_1|, |5S_5 - 3T_5|) \tag{4.6}$$

$$G(i,j)_{LD} = \text{máx}(|5S_3 - 3T_3|, |5S_7 - 3T_7|) \tag{4.7}$$

Las máscaras correspondientes a cada característica direccional se presentan en la Figura 4.2.

La aplicación de las máscaras de Kirsch sobre los patrones genera una nueva representación del patrón en función de la orientación aplicada, que mantiene la misma dimensionalidad que el patrón original. Las Figuras 4.3 y 4.4 muestran un ejemplo de la aplicación de las máscaras de Kirsch sobre una muestra del conjunto de entrenamiento de las bases CENPARMI y MNIST respectivamente.

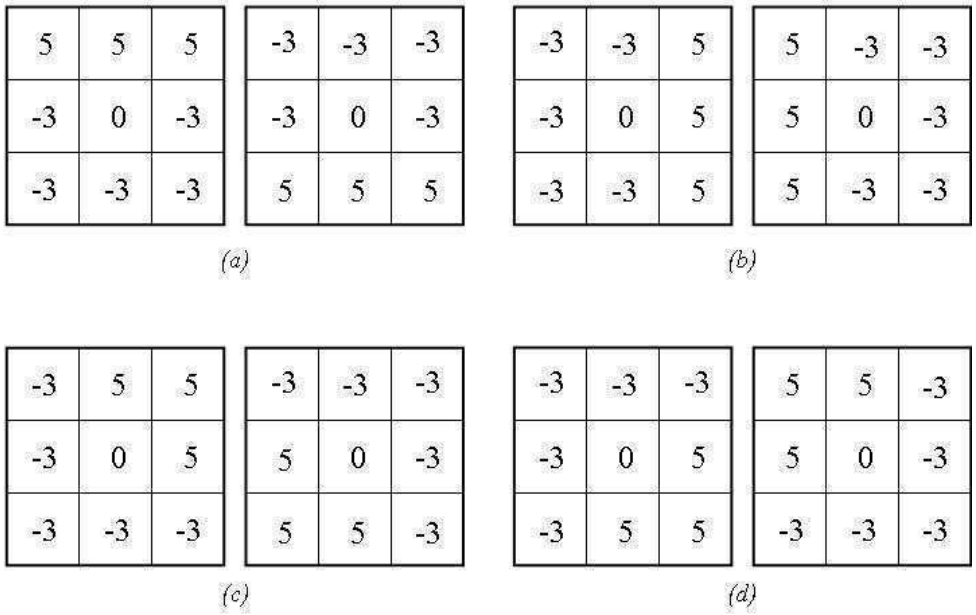


Figura 4.2: Máscaras de Kirsch para extraer las cuatro características direccionales (a) horizontal, (b) vertical, (c) diagonal derecha, (d) diagonal izquierda.

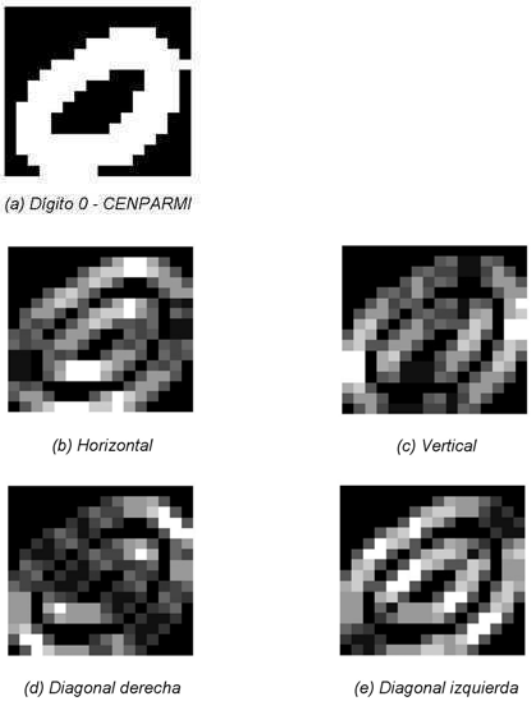


Figura 4.3: Aplicación de las Máscaras de Kirsch sobre (a) un dígito 0 del conjunto de entrenamiento de la base de datos CENPARMI, para extracción de características direccionales: (b) horizontal, (c) vertical, (d) diagonal derecha y (e) izquierda.

4.3. Análisis de Componentes Principales

El análisis de patrones representados con un gran número de variables generalmente requiere una gran cantidad de memoria y poder de cómputo, además de tender a empobrecer el rendimiento del clasificador, ya sea por sobreentrenamiento o disminuyendo su capacidad de generalización [43] [4]. Es por eso que uno de los objetivos en la etapa de preprocesamiento de los datos es reducir su dimensionalidad, de forma tal de llegar a un compromiso entre la calidad de las características representadas y el número de variables utilizado.

El Análisis de Componentes Principales o *Principal Component Analysis* (PCA) es un método estadístico que permite el análisis de datos. En Teoría de las Comunicaciones se lo conoce como la Transformada Karhunen-Loéve [69]. El Análisis de Componentes Principales Lineal es una de las transformaciones más importantes que se utilizan a la hora de reducir la dimensionalidad de los datos en el contexto de la clasificación, debido a los buenos resultados que permite obtener sumado a su simpleza y velocidad a la hora de aplicar el método. Actualmente se utiliza preferentemente como herramienta para el análisis exploratorio de datos y para construir modelos predictivos [43]. Utiliza un procedimiento matemático

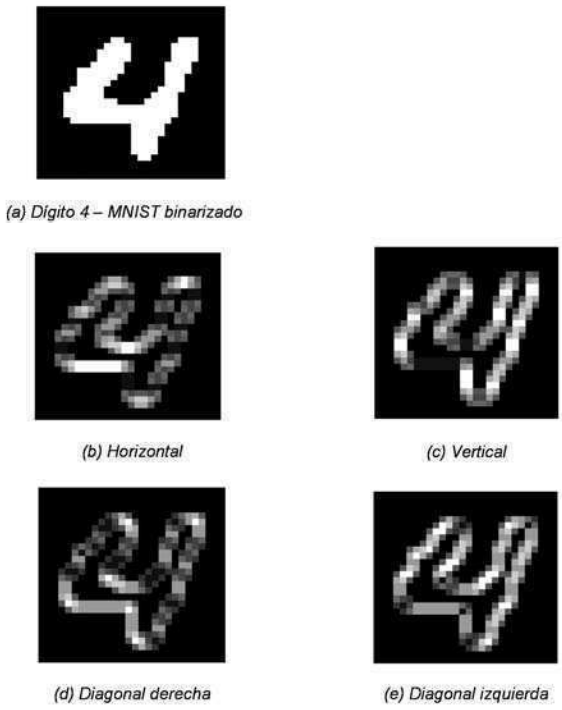


Figura 4.4: Aplicación de las Máscaras de Kirsch sobre (a) un dígito 4 del conjunto de entrenamiento de la base de datos MNIST, para extracción de características direccionales: (b) horizontal, (c) vertical, (d) diagonal derecha y (e) izquierda.

que transforma un número de variables posiblemente correlacionadas a otro sistema de coordenadas denominadas componentes principales, donde las variables están decorrelacionadas. La primer componente principal está asociada a la mayor varianza de los datos.

La técnica de PCA se considera apropiada para ser aplicada al problema de los dígitos manuscritos por varias razones: es un método sencillo que realiza una transformación lineal de los patrones; los componentes del vector de características transformado son estadísticamente independientes; las componentes principales están ordenadas según su importancia, desde la componente asociada a la mayor varianza hasta la asociada a la menor variabilidad de los datos. De esta forma uno podría eliminar las últimas k componentes sin perder demasiada información y utilizando menos variables. El valor de k se determina empíricamente según cada aplicación.

A continuación desarrollaremos los conceptos básicos de PCA.

Sea X un vector aleatorio m-dimensional con media cero,

$$E[X] = 0 \tag{4.8}$$

donde E es la esperanza estadística. Si X tuviera una media distinta de cero, restaríamos la media de X antes de continuar con el análisis.

Sea q el vector unidad también de dimensión m, sobre el cual se proyecta el vector X. Esta proyección se define como el producto interno entre los vectores X y q, como muestra la ecuación 4.9,

$$A = X^T q = q^T X \tag{4.9}$$

sujeto a,

$$\|q\| = \left(q^T q\right)^{1/2} = 1 \tag{4.10}$$

La proyección A es una variable aleatoria con media y varianza relacionadas con los estadísticos del vector aleatorio X. Siendo la media para X igual a cero, la media para A es también cero,

$$E\left[A\right] = q^T E\left[X\right] = 0 \tag{4.11}$$

y la varianza de A se define como

$$\sigma^2 = E\left[A^2\right] = E\left[\left(q^T X\right)\left(X^T q\right)\right] = q^T E\left[XX^T\right] q = q^T R q \tag{4.12}$$

La matriz R de dimensión mxm y simétrica, es la matriz de correlación del vector aleatorio X, formalmente definida como la esperanza del producto externo del vector X consigo mismo. Esta matriz R puede expresarse en términos de sus autovalores y autovectores (teorema espectral), como

$$R = \sum_{i=1}^{m} \lambda_i q_i q_i^T \tag{4.13}$$

siendo λ_i los autovalores de la matriz de correlación R, reales y no negativos. Los autovalores pueden ordenarse en forma decreciente según el subíndice, siendo λ_1 el valor máximo y λ_m el mínimo.

Sea el vector de datos x, denotando una instancia del vector aleatorio X. Teniendo en cuenta la existencia de m soluciones posibles para el vector unitario q, sabemos que hay m proyecciones posibles del vector x. De la ecuación 4.9 tenemos que,

$$a_j = q_j^T x = x^T q_j, j = 1, 2, \ldots, m \tag{4.14}$$

donde los a_j representan las proyecciones de x sobre las direcciones principales dadas por los vectores unidad q_j. Los a_j reciben el nombre de *componentes principales*, y tienen la misma dimensión que el vector de datos x. La fórmula 4.14 se denomina *análisis*.

Para reconstruir el vector de datos original x en forma exacta usando las proyecciones a_j, se procede de la siguiente forma: primero se combina el conjunto de proyecciones $\{a_j \,|j = 1, 2, \ldots, m\}$ en un único vector,

$$a = [a_1, a_2, \ldots, a_m]^T = \left[x^T q_1, x^T q_2, \ldots, x^T q_m\right]^T = Q^T x \tag{4.15}$$

Luego, se premultiplica ambos lados de la ecuación 4.15 por la matriz Q, y teniendo en cuenta que Q está compuesta por los autovectores q_i de la matriz de correlación y que $Q^T = Q^{-1}$ [24], el vector x se reconstruye como

$$x = Qa = \sum_{j=1}^{m} a_j q_j \tag{4.16}$$

La fórmula 4.16 se denomina *síntesis* y representa un cambio de coordenadas tal que un punto x en el espacio de datos es transformado en un punto a en el espacio de características. Notar que los vectores unitarios q constituyen una base del espacio de datos.

Desde la perspectiva del reconocimiento de patrones estadístico, el valor práctico de PCA reside en el hecho que provee una técnica efectiva para reducir la dimensionalidad, jerarquizando la información con la que se representan los datos en función de la varianza asociada con cada una de las coordenadas. De esta forma, en la ecuación 4.16 podrían descartarse aquellas combinaciones lineales con varianza pequeña para mantener sólo aquellos términos de gran varianza.

Sean $\lambda_1, \lambda_2, \ldots, \lambda_l$ los l autovalores de mayor valor asociados con la matriz de correlación R. El vector de datos x puede aproximarse truncando la fórmula 4.16 a l términos, según muestra la ecuación 4.17,

$$\begin{aligned} \widehat{x} &= \textstyle\sum_{j=1}^{l} a_j q_j \\ &= [q_1, q_2, \ldots, q_l] \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_l \end{bmatrix}, \; l \leq m \end{aligned} \tag{4.17}$$

Dado un vector de datos x podemos calcular el conjunto de componentes principales mantenidas en la ecuación 4.17, utilizando la fórmula 4.14 de la siguiente manera:

$$\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_l \end{bmatrix} = \begin{bmatrix} q_1^T \\ q_2^T \\ \vdots \\ q_l^T \end{bmatrix} x,\ l \leq m \tag{4.18}$$

La proyección lineal que muestra la ecuación 4.18 de $\Re^m$ a $\Re^l$, es decir, una correspondencia del espacio de entrada de dimensión m al espacio de características de dimensión l, constituye un *codificador* para la aproximación del vector x, como muestra la Figura 4.5 (a). De la misma manera, la proyección lineal realizada en la fórmula 4.17 de $\Re^l$ a $\Re^m$ (la correspondencia desde el espacio de características hacia el espacio de entrada), representa un *decodificador* para la reconstrucción aproximada del vector de datos original x, como muestra la Figura 4.5 (b). Notar que los autovalores de mayor valor λ_1, λ_2,. . ., λ_l no participan de las ecuaciones 4.17 y 4.18, sino que simplemente determinan el número de componentes principales utilizado en los procesos de codificación y decodificación.

El error de aproximación es un vector e igual a la diferencia entre el vector de datos original x y la aproximación $\widehat{x}$,

$$e = x - \widehat{x} \tag{4.19}$$

El error e es ortogonal al vector de aproximación $\widehat{x}$ [24].

Por otro lado, la varianza total de los m componentes del vector de datos x es,

$$\sum_{j=1}^{m} \sigma_j^2 = \sum_{j=1}^{m} \lambda_j \tag{4.20}$$

donde σ_j^2 es la varianza de la j-ésima componente principal a_j. La varianza total de los l elementos del vector de aproximación $\widehat{x}$ es,

$$\sum_{j=1}^{l} \sigma_j^2 = \sum_{j=1}^{l} \lambda_j \tag{4.21}$$

Finalmente, la varianza total de los $(m - l)$ elementos del vector de error de aproximación e es,

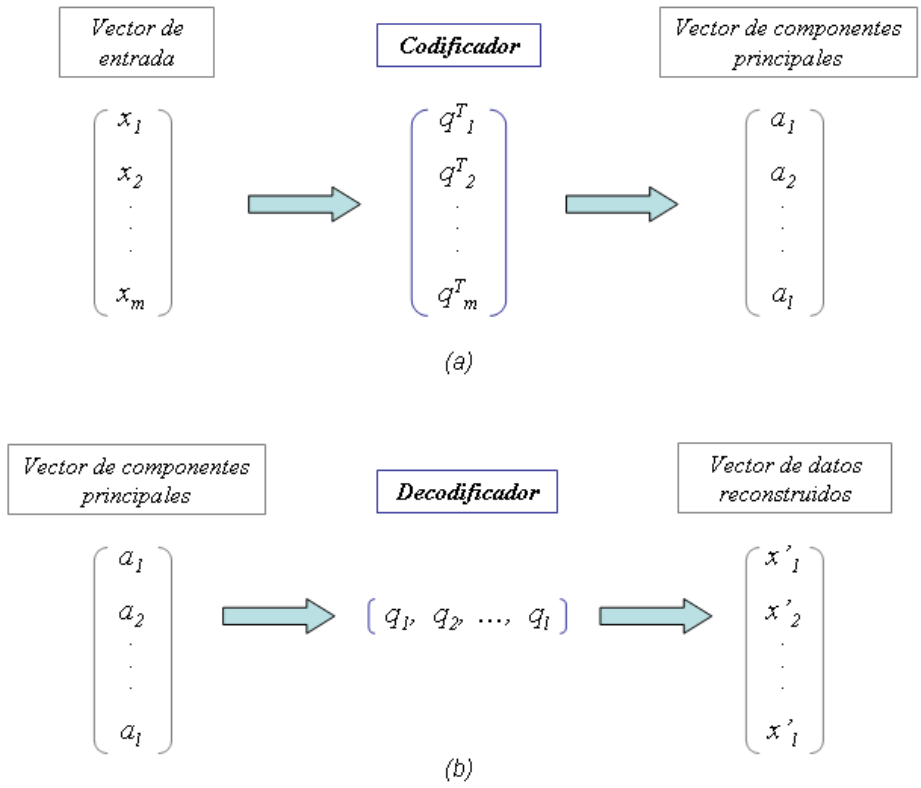


Figura 4.5: Análisis de Componentes Principales: (a) Codificador, (b) Decodificador [24]

$$\sum_{j=l+1}^{m} \sigma_j^2 = \sum_{j=l+1}^{m} \lambda_j \tag{4.22}$$

Los autovalores λ_{l+1},. . ., λ_m tienen asociado los valores más pequeños entre todos los autovalores asociados a la matriz de correlación R, y corresponden a los términos que han sido descartados en la ecuación 4.17 en el proceso de construcción de la aproximación $\widehat{x}$. Cuanto más cerca estén estos autovalores del cero, menor será la pérdida de información en la nueva representación de los datos.

Para resumir diremos que, para realizar una reducción de dimensionalidad sobre un conjunto de datos de entrada obteniendo una nueva representación de los mismos, se deberá calcular los autovalores y autovectores de la matriz de correlación del vector de entrada, para luego proyectar los datos de forma ortogonal en el subespacio generado por los autovectores asociados con los autovalores dominantes.

La Figura 4.6 muestra un ejemplo de la proyección de un conjunto de datos sobre los ejes principales asociados a la distribución.

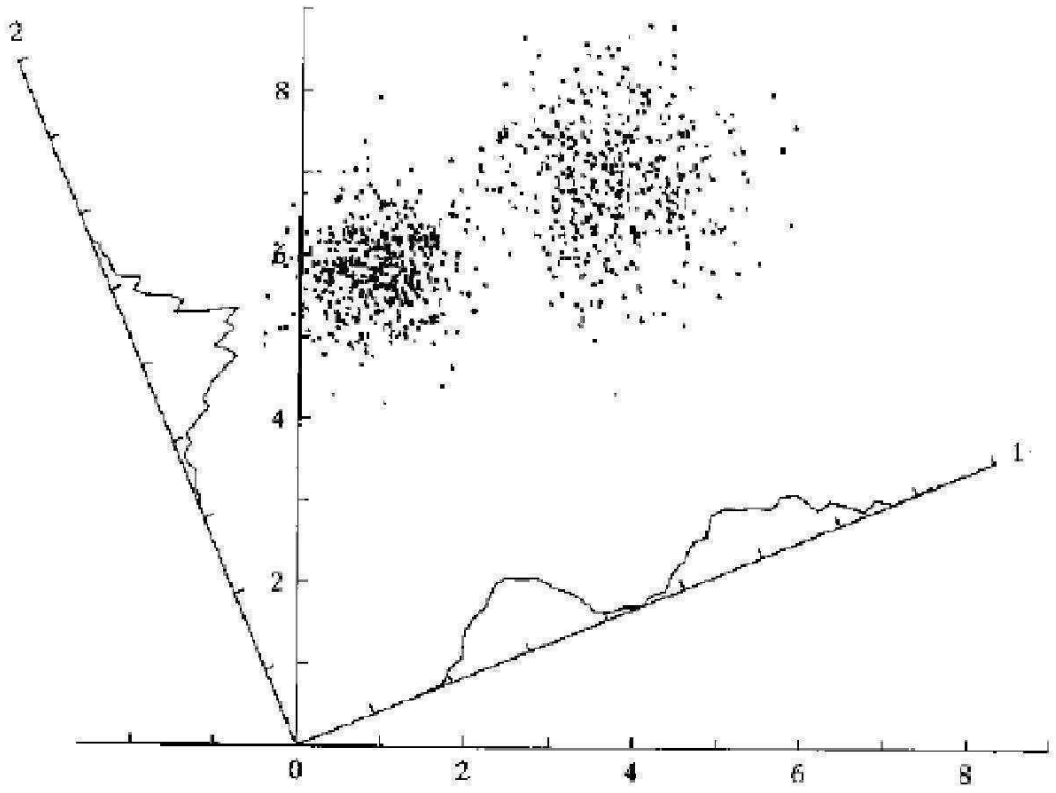


Figura 4.6: Análisis de Componentes Principales: proyección de una nube de puntos de dos dimensiones sobre los ejes 1 y 2. El eje 1 está asociado con la mayor varianza de los datos, permitiendo observar en la proyección de los puntos su carácter bimodal [24].

Existen otros enfoques dentro de PCA, como el denominado *Curvilinear Component Analysis o CCA* [89] o *kernel PCA* [24], ambos métodos no lineales. Una de las dificultades en cuanto su aplicación, reside en los altos tiempos de cómputo necesarios.

4.4. Transformada Wavelet

4.4.1. Introducción

Las wavelets son funciones que satisfacen ciertas propiedades matemáticas y que permiten obtener una representación tiempo-frecuencia de una señal. Una onda es una función oscilatoria periódica del tiempo o espacio. Las onditas o wavelets son ”ondas localizadas”, cuya energía está concentrada en el tiempo o espacio. La Figura 4.7 muestra un ejemplo.

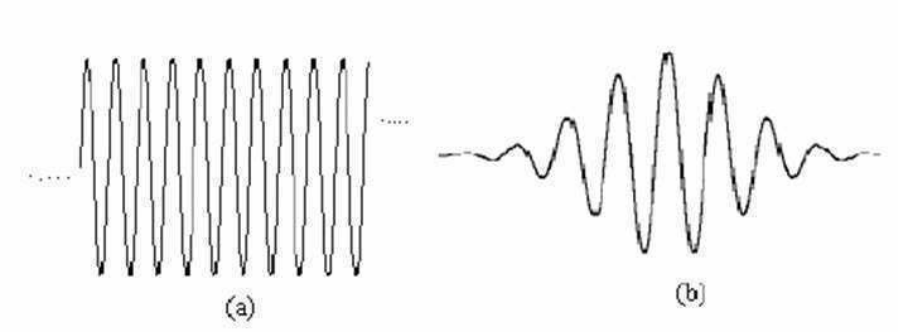


Figura 4.7: Representación de una onda (a) y una wavelet (b).

La representación de señales utilizando una transformada no es una idea nueva. Fue por el 1800 que Joseph Fourier descubrió que superponiendo senos y cosenos podía aproximar otras funciones. La utilidad de la transformada de Fourier reside en su habilidad de analizar una señal representada en el dominio del tiempo por su contenido frecuencial, dado que los coeficientes de Fourier de la función transformada representan la contribución de cada función seno y coseno para cada frecuencia. La transformada discreta de Fourier (DFT) estima la transformada de Fourier en base a un conjunto finito de muestras de la

señal a transformar. La transformada rápida de Fourier (FFT) disminuye la complejidad de cómputo para la aproximación de una función usando la DFT, para muestras de la señal uniformemente espaciadas. Una de las limitaciones de la transformada de Fourier consiste en que la información frecuencial que provee es global, es decir, no es posible determinar dónde la función exhibe cierta característica frecuencial en particular. Para superar este inconveniente se desarrolló la transformada de Fourier con ventana (WFT) también denominada *Short Time Fourier Transform* (STFT), la que permite obtener información de las señales en los dominios de tiempo y frecuencia, además de brindar una solución al problema de representar en forma más precisa una señal no periódica. Sin embargo, la STFT impone un valor fijo de resolución tiempo-frecuencia dentro del plano tiempo-frecuencia, debido a que se utiliza una única ventana para todas las frecuencias [90] [37]. La transformada wavelet, por otro lado, se basa en dilataciones y traslaciones de una función base o tipo $\psi \in L^2(\Re)$ llamada también wavelet madre. Estas bases de funciones tienen corta resolución temporal para altas frecuencias, y una resolución temporal más larga para bajas frecuencias. Esta flexibilidad permite realizar un análisis localizado de la frecuencia de la señal y es una de las características más destacadas. Las transformadas wavelet están asociadas a un conjunto infinito de funciones base posibles. De esta forma, el análisis de señales basado en wavelets permite obtener una mejor representación tiempo-frecuencia de una señal, la cual no es posible realizar con el análisis convencional que plantea la transformada de Fourier y sus variantes [91].

La transformada wavelet ha sido ampliamente estudiada para su aplicación en procesamiento de señales desde 1988 [41]. En la última década, la transformada wavelet ha sido considerada como una herramienta poderosa en un amplio rango de aplicaciones, incluyendo el reconocimiento de patrones [14], procesamiento de imágenes y video [92], análisis numérico, telecomunicaciones, astronomía, acústica, ingeniería nuclear, biomedicina, entre otras. Como ejemplo, se puede mencionar su aplicación a la compresión de señales, donde es utilizada en el estándar para compresión de huellas digitales utilizado por el FBI de Estados Unidos y en el estándar de compresión de imágenes JPEG2000 [93].

En las subsecciones siguientes presentaremos los fundamentos básicos de la teoría de Wavelets, orientado a la comprensión de las herramientas utilizadas en el presente trabajo.

4.4.2. Transformada Wavelet Continua

La wavelet es una función $\psi \in L^2(\Re)$ con media cero,

$$\int_{-\infty}^{\infty} \psi(t)dt = 0. \tag{4.23}$$

normalizada $\|\psi\| = 1$, y centrada alrededor de $t = 0$ [36]. En base a esta función wavelet ψ (wavelet madre), se puede obtener una familia de funciones trasladando y escalando la función tipo:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi(\frac{t-b}{a}) \tag{4.24}$$

donde a, $b \in \Re$, $(a > 0)$. El parámetro a es el factor de escala que permite expandir y comprimir la señal. Escalas más grandes dilatan la señal en el tiempo (bajas frecuencias) permitiendo observar el comportamiento de la señal en forma global. Por otro lado, un factor de escala pequeño comprime la señal (altas frecuencias) brindando información detallada sobre la misma. El parámetro b es el factor de traslación asociado con la ubicación de la función wavelet mientras es desplazada a través de la señal, y corresponde a la información de tiempo en la transformada wavelet.
El hecho que la wavelet esté normalizada asegura que

$$\|\psi_{a,b}(t)\| = \|\psi(t)\| \tag{4.25}$$

Además, la wavelet madre debe cumplir la siguiente condición de admisibilidad:

$$_{\psi} = \int_{-\infty}^{\infty} \frac{|\Psi(\omega)|^2}{\omega} d\omega < \infty \tag{4.26}$$

donde $\Psi(\omega)$ es la transformada de Fourier de $\psi(t)$. La condición de admisibilidad se reduce a [35] [36]:

$$\int_{-\infty}^{\infty} \psi(t)dt = \Psi(0) = 0 \tag{4.27}$$

lo que indica un comportamiento pasabandas de la wavelet.

La Transformada Wavelet Continua (CWT) de una función $f \in L^2(\Re)$, se define como [35]:

$$Wf(a,b) = \int_{-\infty}^{\infty} f(t)\psi_{a,b}(t)dt = \langle f(t), \psi_{a,b}(t) \rangle \tag{4.28}$$

Como resultado del análsis de una señal a través de la CWT, se obtiene un conjunto de coeficientes que indican cuán cercana está la señal con respecto a la función base utilizada.

El hecho de que la CWT cumpla con la condición de admisibilidad permite una reconstrucción exacta de la señal original a partir de los coeficientes wavelet obtenidos y utilizando la fórmula de inversión o antitransformada [35] [36].
En la CWT, el análisis de una señal se realiza utilizando un conjunto de funciones base relacionadas a través de los parámetros de escala y traslación. Uno de los inconvenientes es el hecho de que estos parámetros toman valores continuos, resultando en una representación muy redundante de la CWT e impracticable utilizando una computadora. De esta forma, la evaluación de dichos parámetros se realiza muestreando el plano tiempo-escala, obteniendo un conjunto discreto de funciones base continuas.

La discretización se realiza de la siguiente manera: $a = a_0^j$ y $b = ka_0^j b_0$ para $j, k \in Z$, donde $a_0 > 1$ es el parámetro de escalamiento y $b_0 > 0$ es el parámetro de traslación. Entonces, la familia de wavelets se define como [35],

$$\psi_{j,k}(t) = a_0^{-j/2}\psi(a_0^{-j}t - kb_0) \tag{4.29}$$

y la descomposición wavelet de una función f es,

$$f(t) = \sum_j \sum_k D_f(j,k)\psi_{j,k}(t) \tag{4.30}$$

donde el conjunto de coeficientes $D_f(j,k)$ constituye la transformada wavelet discretizada de la función $f(t)$.

La implementación de esta versión discretizada de la CWT puede resultar computacionalmente costosa en tiempo y recursos, dependiendo de la resolución requerida. Como ejemplo, mencionaremos la función

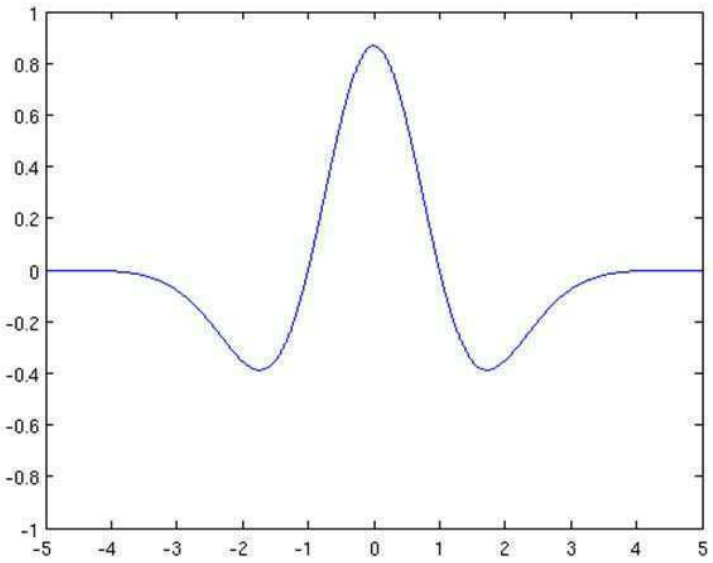


Figura 4.8: Wavelet Mexican Hat $\psi(x) = \frac{2}{\sqrt{3}}\pi^{-1/4}(1-x^2)e^{-x^2/2}$.

Mexican Hat (ver Figura 4.8) que es la derivada segunda de la función Gaussiana $e^{-x^2/2}$, y que no tiene función de escala asociada.

La función Mexican Hat es muy utilizada en distintas aplicaciones, como por ejemplo, en la representación de características de señales en el área de Reconocimiento de Patrones [42].

4.4.3. Transformada Wavelet Discreta

La Transformada Wavelet Discreta (DWT) se basa en la técnica de codificación subbanda y constituye una variante de la Transformada Wavelet que es fácil de implementar y que reduce los tiempos computacionales y recursos requeridos.

Análisis en Multi-Resolución utilizando Bancos de Filtros

La utilidad práctica de la DWT viene dada por sus propiedades de análisis en Multi-Resolución (MRA) y de reconstrucción sin pérdida utilizando estructuras de bancos de filtros [91].
Una secuencia $V_{j\,j\in Z}$ de subespacios cerrados de $L^2(\Re)$ es una *aproximación en multiresolución* si se cumplen las siguientes características [36]:

1. $\forall(j,k) \in Z^2, f(t) \in V_j \Leftrightarrow f(t-2^j k) \in V_j$

2. $\forall(j) \in Z, V_{j+1} \subset V_j$

3. $\forall(j) \in Z, f(t) \in V_j \Leftrightarrow f(\frac{t}{2}) \in V_{j+1}$

4. $\lim_{j\to+\infty} V_j = \bigcap_{j=-\infty}^{+\infty} V_j = \{0\}$

5. $\lim_{j\to-\infty} V_j =$ Clausura $(\bigcup_{j=-\infty}^{+\infty} V_j) = L^2(\Re)$

6. Existe θ tal que $\theta(t-n)_{n\in Z}$ es una base de Riesz de V_0.

Dada una secuencia cualquiera de subespacios anidados que cumpla con las propiedades definidas anteriormente para la aproximación en multiresolución, existe una única función $\varphi \in L^2(\Re)$ asociada, llamada *función de escala*, tal que:

$$\forall(j,n) \in Z^2, \varphi_{j,n}(x) = 2^{-j/2}\varphi(2^{-j}x - n) \tag{4.31}$$

siendo $\{\varphi_{j,n}; n \in Z\}$ una base ortonormal para $V_j, \forall(j) \in Z$.

Para cada $j \in Z$ se define W_j como el complemento ortogonal de V_j, ambos en V_{j-1} y siendo $\oplus$ el operador suma directa de dos espacios vectoriales, es decir,

$$V_{j-1} = V_j \oplus W_j \tag{4.32}$$

y

$$W_j \perp W_{j'}, j \neq j' \tag{4.33}$$

Entonces, existe una función wavelet o *wavelet madre* ψ, tal que $\{\psi_{j,k}; j, k \in Z\}$ es una base ortonormal de $L^2(\Re)$, y

$$\psi_{j,k}(x) = 2^{-j/2}\psi(2^{-j}x - k) \tag{4.34}$$

donde para un $j \in Z$ fijo, $\{\psi_{j,k}; k \in Z\}$ constituye una base ortonormal para W_j. Esto último es equivalente a decir que $\forall f \in L^2(\Re)$

$$P_{j-1}f = P_j f + \sum_{k \in Z} \langle f, \psi_{j,k} \rangle \psi_{j,k} \tag{4.35}$$

siendo P_j la proyección ortogonal en V_j. Es decir, la proyección ortogonal de la función f en V_{j-1} puede descomponerse como la suma de las proyecciones ortogonales sobre V_j y W_j, o sea, como la aproximación de la función y la sumatoria de los detalles aportados por la wavelet, correspondientes a los subespacios anidados y de menor resolución.

Transformada Wavelet Ortogonal Rápida

La Transformada Wavelet Ortogonal Rápida (FWT) descompone cada aproximación $P_{V_j} f$ de la función f, en aproximaciones de menor resolución $P_{V_{j+1}} f$ más los coeficientes wavelet aportados por la proyección $P_{W_{j+1}} f$. En el otro sentido, la reconstrucción a partir de los coeficientes wavelet obtiene cada $P_{V_j} f$ a partir de $P_{V_{j+1}} f$ y $P_{W_{j+1}} f$.
Dado que $\{\varphi_{j,n}; j, n \in Z\}$ y $\{\psi_{j,n}; j, n \in Z\}$ son bases ortonormales de V_j y W_j, las proyecciones en estos subespacios están caracterizadas por:

$$a_j[n] = \langle f, \varphi_{j,n} \rangle, d_j[n] = \langle f, \psi_{j,n} \rangle \tag{4.36}$$

donde $a_j[n]$ y $d_j[n]$ corresponden a los coeficientes de aproximación y detalle respectivamente. Dichos

coeficientes se calculan en una cascada de convoluciones y submuestreos, según el algoritmo de Mallat [36]:

$$a_{j+1}[p] = \sum_{n=-\infty}^{+\infty} h[n-2p]a_j[n] = a_j * \overline{h}[2p] \tag{4.37}$$

$$d_{j+1}[p] = \sum_{n=-\infty}^{+\infty} g[n-2p]a_j[n] = a_j * \overline{g}[2p] \tag{4.38}$$

siendo $\overline{x}[n] = x[-n]$, y $\overline{h}$, $\overline{g}$ filtros de bajas y altas frecuencias respectivamente.

Las ecuaciones 4.37 y 4.38 indican la etapa de descomposición de la señal (ver Figura 4.9). En cada

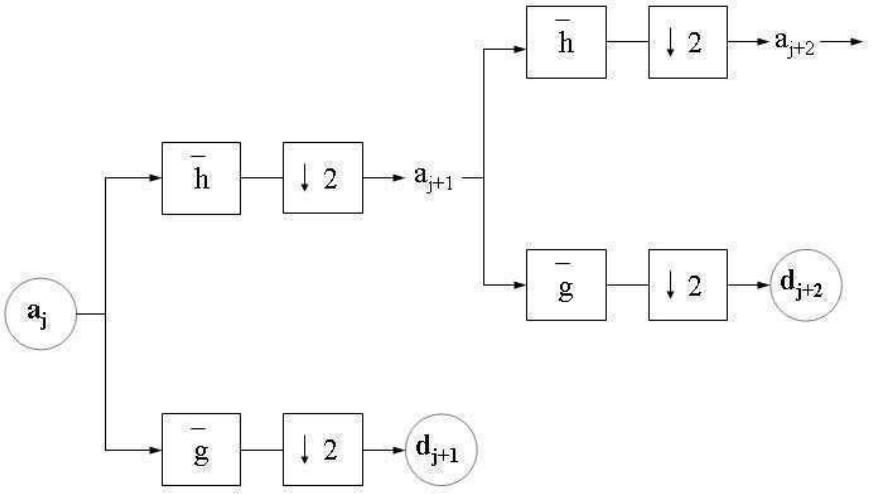


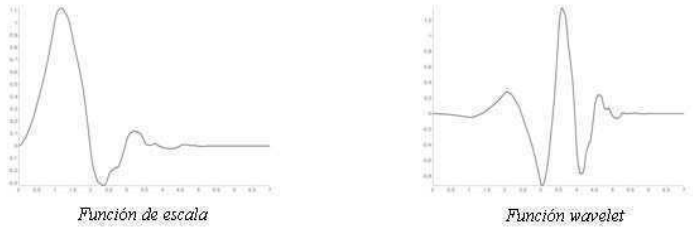
Figura 4.9: Descomposición usando la FWT, aplicando una cascada de filtros $\overline{h}$ y $\overline{g}$ seguidos por una operación de submuestreo o decimación de a 2 ($\downarrow 2$) donde a la señal se le eliminan los componentes pares.

nivel, el filtro pasa altos genera información de detalle d_j, mientras que el filtro pasa bajos asociado con la función de escala, produce aproximaciones suavizadas a_j de la señal.

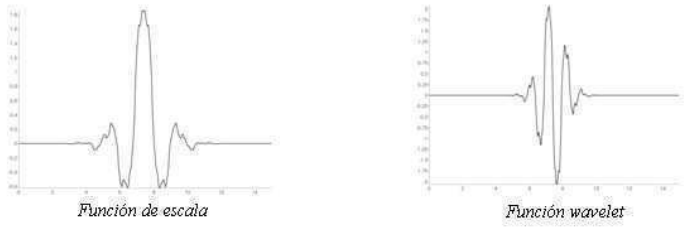
La implementación de la FWT requiere sólo $O(N)$ operaciones para señales de tamaño N, permitiendo además la representación de la señal en forma no redundante y la reconstrucción exacta (sin pérdida) de la misma.

Este algoritmo de bancos de filtros con reconstrucción perfecta también puede aplicarse a la Transformada Wavelet Biortogonal Rápida. Los coeficientes wavelet se calculan mediante sucesivas convoluciones con filtros $\overline{h}$ y $\overline{g}$, mientras que para la reconstrucción de la señal se utilizan los filtros duales $\widetilde{h}$ y $\widetilde{g}$. Si la señal incluye N muestras distintas de cero, su representación basada en la wavelet biortogonal se calcula con $O(N)$ operaciones [36].

La Figura 4.10 muestra una wavelet ortogonal y otra biortogonal, ambas ampliamente difundidas.



(a) Transformada Wavelet Ortogonal Daubechies 4



(b) Transformada Wavelet Biortogonal CDF 3/7 para la descomposición

Figura 4.10: Funciones de ejemplo de la Transformada Wavelet (a) ortogonal (b) biortogonal.

Transformada Wavelet Discreta en 2 Dimensiones (2-D DWT)

La estructura de bancos de filtros vista anteriormente es la implementación más simple de la DWT en una dimensión. El procesamiento de imágenes digitales requiere de la implementación de la transformada wavelet en dos dimensiones, también denominada multidimensional. La FWT en dos dimensiones es una extensión de la FWT unidimensional aplicada a filas y luego a columnas. Sea $\psi(x)$ la wavelet unidimensional asociada a la función de escala unidimensional $\varphi(x)$, entonces la función de escala en 2D está dada por,

$$\varphi(x,y)_{LL} = \varphi(x)\varphi(y) \tag{4.39}$$

y las tres wavelets 2D están definidas por,

$$\psi(x,y)_{LH} = \varphi(x)\psi(y) \tag{4.40}$$

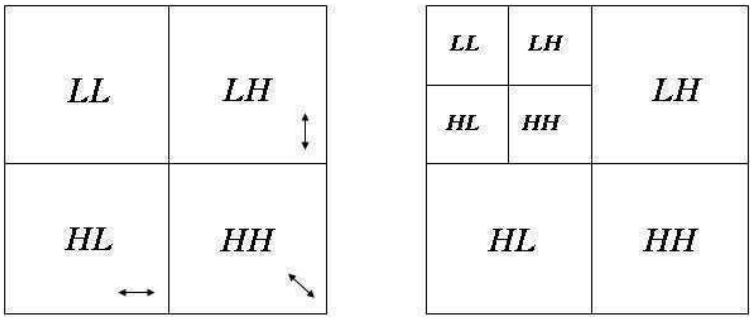
$$\psi(x,y)_{HL} = \psi(x)\varphi(y) \tag{4.41}$$

$$\psi(x,y)_{HH} = \psi(x)\psi(y) \tag{4.42}$$

donde LL representa las frecuencias más bajas (información global), LH representa las altas frecuencias verticales (detalles horizontales), HL las altas frecuencias horizontales (detalles verticales), y HH representa las altas frecuencias en ambas diagonales (detalles diagonales).

La aplicación de un paso de la transformada sobre la imagen original, produce una subbanda de aproximación LL que corresponde a la imagen suavizada, y tres subbandas de detalle HL, LH y HH. El siguiente paso de la transformada se aplica sobre la subbanda de aproximación, dando como resultado otras cuatro subbandas, como muestra la Figura 4.11. Es decir, cada paso en la descomposición representa a la subbanda de aproximación del nivel i en cuatro subbandas en el nivel $i+1$, cada una de las cuales tendrá tamaño un cuarto con respecto a la imagen o subbanda que la originó.

Aunque la DWT estándar es una herramienta poderosa, presenta algunas limitaciones, como por ejemplo ser sensible a traslaciones y detectar sólo algunas características en cuanto a direccionalidad. Sin embargo, creemos apropiada la aplicación de este tipo de wavelet para el problema del reconocimiento de dígitos manuscritos, ya que las imágenes que estamos procesando están normalizadas en tamaño y centradas, y la direccionalidad de un número aunque sea manuscrito no presenta demasiadas variantes.



(a) Nivel 1 o aplicación de un paso DWT

(b) Nivel 2 o aplicación de dos pasos DWT

Figura 4.11: Descomposición multinivel de una imagen con la 2D DWT: (a) Nivel 1, (b) Nivel 2.

Capítulo 5

Construcción de un descriptor basado en la Transformada Wavelet y PCA

El objetivo de este Capítulo es presentar buenos descriptores que permitan mejorar los resultados en la clasificación de dígitos manuscritos, y a su vez permitan apreciar las bondades del método de clasificiación propuesto en este trabajo de tesis. En función de este objetivo se han aplicado técnicas de multirresolución y de Análisis de Componentes Principales, presentadas en el Capítulo 4.

5.1. Introducción

Como ya hemos mencionado en el Capítulo 4, la Transformada Wavelet es una herramienta especialmente utilizada para localizar información espacial y frecuencial en el procesamiento de imágenes y, en particular, para la extracción de características de patrones orientada a la clasificación. Numerosos trabajos han utilizado esta técnica en diversas áreas [14] [94] [41]. En particular, para el problema del reconocimiento de dígitos manuscritos, diferentes enfoques han sido publicados. Por ejemplo, en [37] se aplica una wavelet ortogonal discreta unidimensional sobre el contorno de los dígitos, construyendo un descriptor con las bandas de aproximación de la transformada. El objetivo es lograr que las variaciones de forma causadas por los diferentes estilos de escritura no afecten la clasificación. En [39] se presenta un descriptor que utiliza multiwavelets ortonormales sobre el contorno normalizado de cada dígito, hasta el tercer nivel de detalle, de forma de lograr una representación de cada patrón desde un detalle más fino a una aproximación más suavizada del mismo. En [42] se utiliza la CWT Mexican Hat discretizada para extraer una versión más pequeña de cada dígito y el Gradiente Wavelet para conformar un vector complementario con características de orientación, gradiente y curvatura a diferentes escalas. En [38] se

utiliza la familia de wavelets biortogonales *CDF*, en su versión bidimensional, obteniendo un descriptor con las cuatro subbandas del primer nivel de resolución de la transformada, normalizadas.

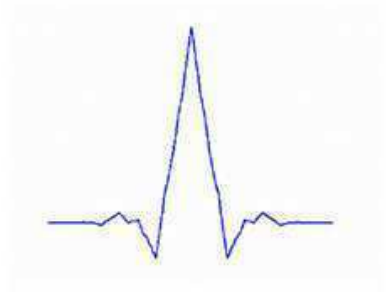
Sabemos que el preprocesamiento es una etapa fundamental en el proceso general de clasificación, ya que en éste se definen qué características serán relevantes a la hora de discriminar entre patrones de distintas clases. Nuestro objetivo es definir un descriptor basado en wavelets eficiente, es decir, con el cual se pueda obtener un porcentaje elevado de reconocimiento y que permita reducir la dimensionalidad de los datos a clasificar. Esta última es una característica de gran importancia a la hora de aplicar métodos de clasificación, dado que influye en su rendimiento y también permite disminuír el costo computacional cuando debemos trabajar con grandes bases de datos. Muchas veces la reducción de la dimensionalidad posibilita la aplicación de ciertos métodos computacionales que de otra manera y según con los recursos disponibles, haría impracticable o inservible el tratamiento de los datos. Finalmente, como orientamos el trabajo a la correcta clasificacion de los patrones, nos interesa la representación de los mismos más que su reconstrucción en función de los coeficientes obtenidos con la Transformada Wavelet.

5.2. Descriptores basados en la Transformada Wavelet *CDF 9/7*

Para el presente trabajo, y respondiendo a los objetivos mencionados, decidimos utilizar la wavelet biortogonal bidimensional *Cohen-Daubechies-Feauveau (CDF) 9/7* ya que ésta es una wavelet especialmente efectiva en la línea de investigación trazada en este trabajo de tesis [95]. Su efectividad en la compresión de imágenes queda demostrada por su utilización en el estándar de compresión JPEG2000, y para la compresión de huellas digitales por el FBI de Estados Unidos [93]. La Figura 5.1 muestra las funciones de escala y wavelet y el valor de los coeficientes de los filtros asociados, para la transformada CDF 9/7 en la descomposición.

Hemos aplicado la CDF 9/7 hasta un segundo nivel de resolución sobre los patrones de las bases de CENPARMI y MNIST descriptos en el Apéndice A. La aplicación de uno y dos pasos de la transformada sobre la imagen de un dígito manuscrito se muestra en la Figura 5.2, para muestras de ambas bases de datos. Puede observarse que en el segundo nivel la forma del dígito pierde detalle, con lo cual decidimos, por el momento, no considerar más niveles en la aplicación de la transformada.

Para la conformación del descriptor hemos considerado distintas variantes: las bandas de aproximación (LL) de primer y segundo nivel de resolución presentan una imagen suavizada del patrón, que preserva su forma y reduce la dimensionalidad en un cuarto de la original en un primer nivel, y dieciseis veces en un segundo nivel, donde la imagen es más burda. La subbanda de altas frecuencias HH en un primer nivel, registra los cambios bruscos en los bordes de la imagen (detalles diagonales), que creemos



(a) Función de escala

(b) Función wavelet

Función de escala	*Función wavelet*
[0.0378,	[0,
-0.0238,	0.0645,
-0.1106,	-0.0407,
0.3774,	-0.4181,
0.8527,	0.7885,
0.3774,	-0.4181,
-0.1106,	-0.0407,
-0.0238,	0.0645,
0.0378]	0]

(c) Filtros

Figura 5.1: CDF 9/7 (a) Función de escala, (b) Función wavelet, (c) Filtros asociados para la descomposición de la señal.

podrían aportar a la hora de detectar diferencias entre los patrones, como así también las subbandas LH y HL. Las subbandas de detalle de segundo nivel, aportan características de altas frecuencias en sentido vertical, horizontal y diagonal sobre la versión suavizada del dígito, lo que puede marcar características en la estructura básica del patrón, de utilidad en el proceso de clasificación.

Otra cuestión a tener en cuenta para la definición del descriptor es establecer qué valores son representativos en las subbandas obtenidas. De esta manera trabajamos con algunos descriptores umbralados y binarizados. Para esto, calculamos algunos estadísticos sobre los valores de cada subbanda, como la media, mediana, varianza y desviación estándar. La utilización de la media como umbral, es lo que nos dio mejores resultados en cuanto a descriptores umbralados. De esta forma descartamos valores muy pequeños considerados no representativos y luego binarizamos los descriptores. La Tabla 5.1 presenta los descriptores considerados, mientras que en las Figuras 5.3 y 5.4 se observan los distintos preprocesamientos sobre muestras de dígitos de ambas bases de datos.

Tabla 5.1: Descriptores utilizando la transformada wavelet CDF 9/7.

		Nombre	Descripción
		Sin Preprocesar	Imagen binarizada normalizada en tamaño
I	(a)	LL1	Aproximación Nivel 1
	(b)	LL1uM	Aprox. Nivel 1 - Umbral: Media - Binarizado
II	(a)	LL1LH1	Aprox.+LH Nivel 1
	(b)	LL1LH1uM	Aprox.+LH Nivel 1 - Umbral: Media - Binarizado
III	(a)	LL1HL1	Aprox.+HL Nivel 1
	(b)	LL1HL1uM	Aprox.+HL Nivel 1 - Umbral: Media - Binarizado
IV	(a)	LL1HH1	Aprox.+HH Nivel 1
	(b)	LL1HH1uM	Aprox.+HH Nivel 1 - Umbral: Media - Binarizado
V	(a)	T2	Transformada Nivel 2 (4 subbandas)
	(b)	T2uM	Transformada Nivel 2 - Umbral: Media - Binarizado
VI	(a)	LL1T2	Aprox. Nivel 1 + Transformada Nivel 2
	(b)	LL1T2uM	Aprox. Nivel 1 + Transformada Nivel 2 - Umbral: Media - Binarizado

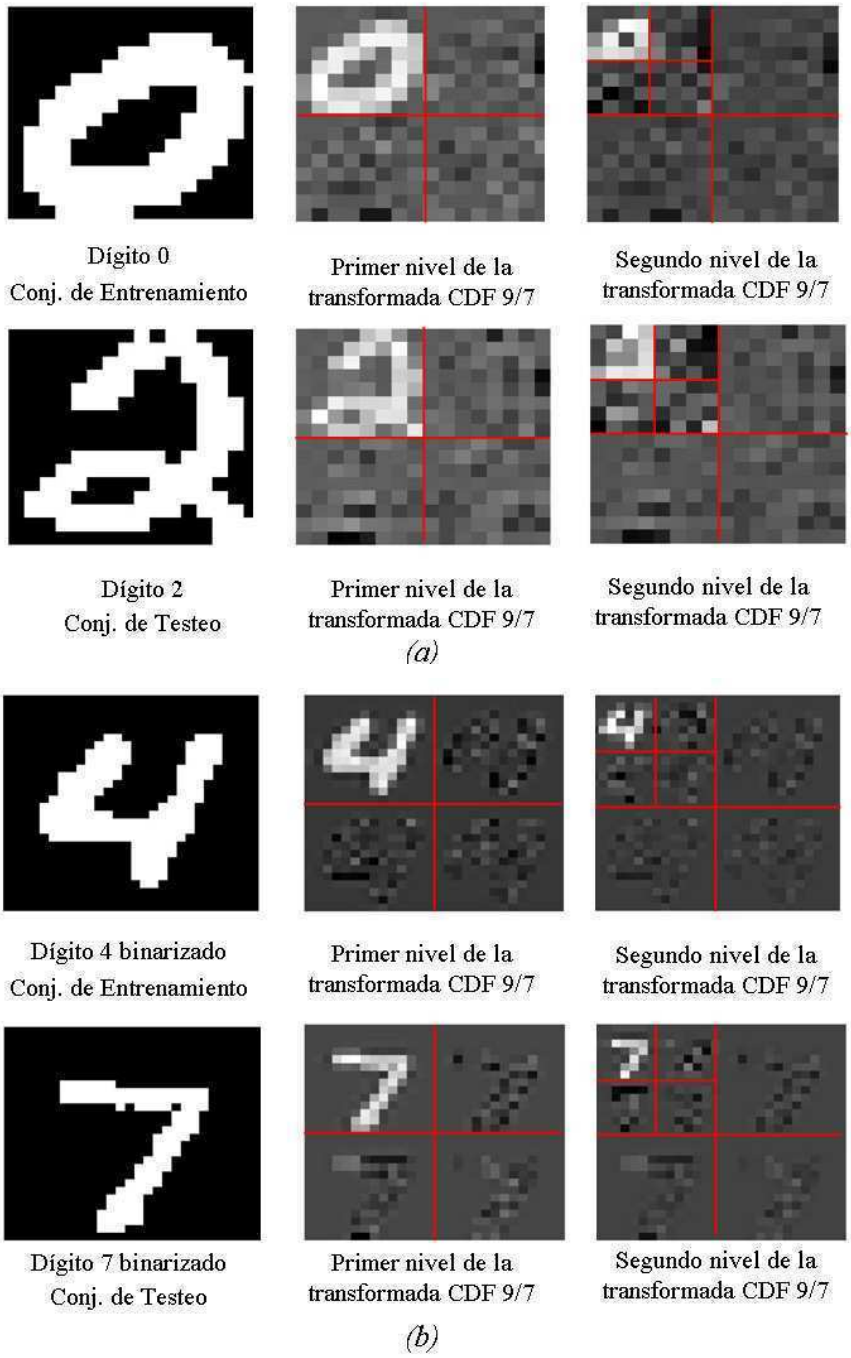


Figura 5.2: Aplicación de la transformada wavelet CDF 9/7 hasta el segundo nivel de resolución para muestras de las bases de datos (a) CENPARMI y (b) MNIST.

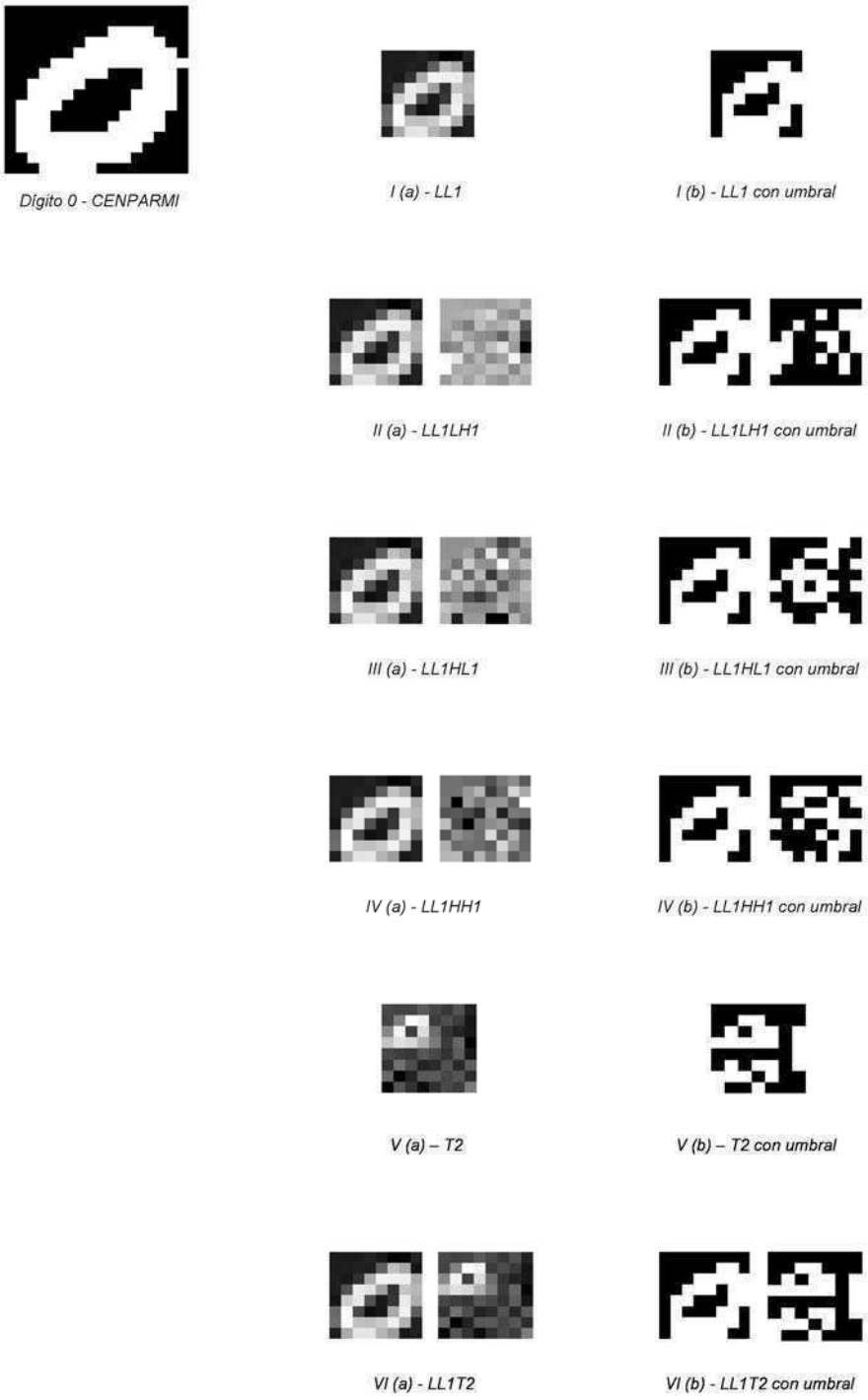


Figura 5.3: Ejemplos de preprocesamiento utilizando la CDF 9/7 según la Tabla 5.1, para dígitos de la base CENPARMI. La dimensión del dígito de muestra es de 256 mientras que la de los descriptores compuestos por una sola subbanda y por dos subbandas tienen dimensión 64 y 128 respectivamente.

Figura 5.4: Ejemplos de preprocesamiento utilizando la CDF 9/7 según la Tabla 5.1, para dígitos de la base MNIST binarizada. La dimensión del dígito de muestra es de 784 mientras que la de los descriptores compuestos por una sola subbanda y por dos subbandas tienen dimensión 196 y 392 respectivamente.

Para evaluar el rendimiento de los descriptores definidos se utilizó la técnica del perceptrón multicapa (MLP) entrenado con el algoritmo de Back-Propagation con momentum y velocidad de aprendizaje adaptativa según lo descripto en el Capítulo 3. La Tabla 5.2 muestra los porcentajes de patrones del conjunto de testeo correctamente clasificados, junto con la dimensionalidad de cada descriptor para la base CENPARMI (ver Apéndice A).

Tabla 5.2: Porcentajes de Reconocimiento sobre conjunto de Testeo CENPARMI usando MLP - *: mejor resultado.

		Descriptor	Dimensión	MLP (% Reconocidos)	
		Sin Preprocesar	256	89.05	
I	(a)	LL1	64	*91.65*	*
	(b)	LL1uM	64	87.95	
II	(a)	LL1LH1	128	91.35	
	(b)	LL1LH1uM	128	86.15	
III	(a)	LL1HL1	128	90.75	
	(b)	LL1HL1uM	128	87.80	
IV	(a)	LL1HH1	128	90.85	
	(b)	LL1HH1uM	128	85.55	
V	(a)	T2	64	*92.15*	*
	(b)	T2uM	64	76.75	
VI	(a)	LL1T2	128	*92.25*	*
	(b)	LL1T2uM	128	88.25	

La Figura 5.5 muestra un gráfico comparativo de los mejores porcentajes de reconocimiento para los resultados de la Tabla 5.2. Notar que los porcentajes más altos corresponden a descriptores sin umbralar, como los descriptores I(a) LL1, V(a) T2 y VI(a) LL1T2.

Para la base de datos MNIST (ver Apéndice A) hemos extraído 15000 patrones del conjunto de entrenamiento para la etapa de aprendizaje, mientras que el conjunto de testeo fue utilizado en su totalidad. Además hemos utilizado la base binarizada, como se explicó en la Sección 3.4. La Tabla 5.3 presenta los porcentajes de patrones del conjunto de testeo correctamente clasificados, junto con la dimensionalidad de cada descriptor.

La Figura 5.6 presenta un gráfico comparativo de los mejores porcentajes de reconocimiento para los resultados de la Tabla 5.3 para MNIST binarizada, utilizando un clasificador del tipo MLP. Notar que los porcentajes más altos también corresponden a descriptores sin umbralar, como los descriptores I(a) LL1, V(a) T2 y VI(a) LL1T2.

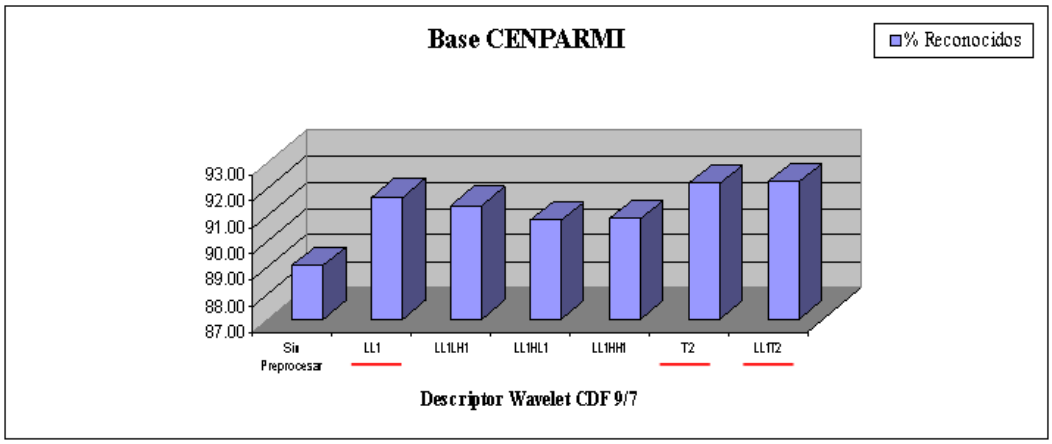


Figura 5.5: Comparación de los resultados de la Tabla 5.2 para los descriptores sin umbralar para la base CENPARMI utilizando MLP.

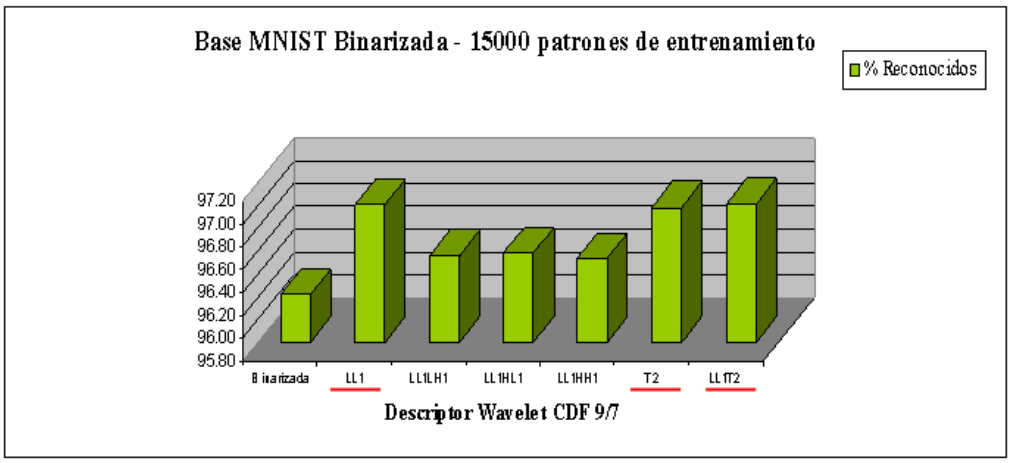


Figura 5.6: Comparación de los resultados de la Tabla 5.3 para los descriptores sin umbralar para la base MNIST (15000 patrones) utilizando MLP.

Tabla 5.3: Porcentajes de Reconocimiento sobre MNIST binarizada utilizando un conjunto de entrenamiento de 15000 patrones y todo el conjunto de Testeo, usando MLP - *: mejor resultado.

		Descriptor	Dimensión	MLP (% Reconocidos)	
		Binarizada	784	96.22	
I	(a)	LL1	196	*97.00*	*
	(b)	LL1uM	196	94.64	
II	(a)	LL1LH1	392	96.56	
	(b)	LL1LH1uM	392	93.32	
III	(a)	LL1HL1	392	96.58	
	(b)	LL1HL1uM	392	93.96	
IV	(a)	LL1HH1	392	96.54	
	(b)	LL1HH1uM	392	93.77	
V	(a)	T2	196	*96.98*	*
	(b)	T2uM	196	89.42	
VI	(a)	LL1T2	392	*97.01*	*
	(b)	LL1T2uM	392	94.21	

Sobre los descriptores con mejor rendimiento para ambas bases de datos se aplicó otra técnica de clasificación muy difundida y eficiente, las Máquinas de Soporte Vectorial o SVM, descriptas en el Capítulo 3. En términos generales, los resultados presentados en las tablas 5.4 y 5.5 mejoran los obtenidos con la técnica de MLP presentados en las tablas 5.2 y 5.3.

La utilización del MLP ha permitido comparar el rendimiento de los distintos descriptores. Hemos utilizado el criterio por el cual, en base a los resultados obtenidos con este método de clasificación, elegimos los mejores descriptores y luego entrenamos clasificadores aplicando SVM. De esta forma evitamos realizar el complicado ajuste de parámetros (asociado con este último método y con los datos utilizados) para todos los experimentos.

Por lo observado para los clasificadores MLP y SVM, las cuatro subbandas del segundo nivel de la transformada CDF 9/7 logran describir los dígitos en forma eficiente logrando una reducción en la dimensionalidad del 75 %. Los resultados mejoran con el agregado de la aproximación de primer nivel, donde la imagen suavizada es menos burda que la de segundo nivel. Creemos que la aproximacion de primer nivel hace su aporte en cuanto a la estructura básica de cada dígito y esto se refleja en los resultados. En este caso la reducción de la dimensionalidad con respecto a la imagen sin preprocesar es del 50 %. Por otro lado, la utilización de la subbanda de aproximación de primer nivel como descriptor,

Tabla 5.4: Porcentajes de Reconocimiento sobre conjunto de Testeo CENPARMI usando SVM multiclase con kernel Gaussiano con $\sigma = 5{,}50$, $4{,}75$ y $5{,}60$ para los mejores resultados obtenidos con los descriptores I(a), V(a) y VI(a) respectivamente.

		Descriptor	Dimensión	MLP (% Reconocidos)	SVM (% Reconocidos)
		Sin preprocesar	256	89.05	85.00
I	(a)	LL1	64	91.65	94.20
V	(a)	T2	64	92.15	94.45
VI	(a)	LL1T2	128	92.25	*94.75*

Tabla 5.5: Porcentajes de Reconocimiento sobre conjunto de Testeo MNIST (para MNIST binarizada con 15000 patrones de entrenamiento) usando SVM multiclase con kernel Gaussiano con $\sigma = 15{,}00$, $15{,}00$ y $21{,}00$ para los mejores resultados obtenidos con los descriptores I(a), V(a) y VI(a) respectivamente.

		Descriptor	Dimensión	MLP (% Reconocidos)	SVM (% Reconocidos)
		Binarizado	784	96.22	97.33
I	(a)	LL1	196	97.00	97.59
V	(a)	T2	196	96.98	97.54
VI	(a)	LL1T2	392	97.01	*97.60*

también es una buena opción dado que produce un buen rendimiento y disminuye la dimensionalidad en un 75 %.

La reducción del tamaño del descriptor es una característica muy importante, sobre todo para las bases de datos de gran volumen y alta dimensionalidad, como es el caso de MNIST. Esta última contiene patrones de tamaño 28x28, lo que implicaría un descriptor para el patrón sin preprocesar de 784 (3 veces más que las imágenes de CENPARMI). Además la base cuenta con 70000 patrones (casi 12 veces más que la cantidad de patrones utilizados para entrenar y testear con la base de CENPARMI).

5.2.1. Aplicación de PCA sobre descriptores basados en la TW (TW-PCA)

Como vimos en la Sección 4.3 la aplicación de PCA permite mantener las coordenadas que retienen la mayor varianza de los datos, eliminando las que se consideran no constituyen un aporte. Esto permite mejorar la calidad de la representación disminuyendo la dimensionalidad de la misma. El resultado es el aumento o mantenimiento del rendimiento en la clasificación junto con la disminución del costo computacional a la hora de entrenar y clasificar. En algunas ocasiones, directamente este punto permite definir si una técnica podrá ser aplicada sobre los datos o no, en función de los recursos computacionales con los que se cuenta y de los tiempos de procesamiento requeridos.

Por las características ya descriptas, se aplicó PCA sobre los descriptores basados en la TW que mejor han funcionado, vistos en la Sección anterior: LL1, T2 y LL1T2, es decir, trabajamos con la aproximada de primer nivel, con la transformada de segundo nivel, y con el descriptor conformado por ambas, para la wavelet CDF 9/7.

No existe una regla para determinar a priori la cantidad óptima de componentes principales a usar. Por esta razón nos hemos guiado por el porcentaje de varianza que retiene un conjunto dado de componentes y por el porcentaje de reducción de la dimensionalidad del descriptor. Es decir, nos interesó reducir considerablemente la dimensionalidad reteniendo la mayor cantidad de información. De esta forma, observando los resultados sobre los conjuntos de datos fue posible adoptar criterios donde prevaleció la reducción del descriptor en al menos un 50 % y la retención de más del 80 % de la varianza total.

En la Tabla 5.6 se presenta la experimentación asociada a CENPARMI.

Observamos que en el caso Sin Preprocesar, la aplicación de PCA no sólo reduce la dimensionalidad en un 50 y hasta un 75 %, sino que a su vez aumenta el porcentaje de patrones correctamente clasificados usando MLP. En el caso de utilizar los descriptores basados en la TW con PCA el rendimiento mejora para LL1T2 reduciendo el tamaño del descriptor en un 50 %. Para los descriptores que ya tienen baja dimensionalidad como es el caso de LL1 y T2, la aplicación de PCA no mejora los resultados.

Para el caso MNIST binarizada, los valores se presentan en la Tabla 5.7 para 15000 patrones de entrenamiento. A la hora de determinar la cantidad de componentes principales, fue posible establecer el siguiente criterio: reducir la dimensionalidad en al menos un 50 % reteniendo más del 90 % de la varianza.

La aplicación de PCA sobre los descriptores seleccionados basados en la TW permite mejorar los resultados con respecto a la base binarizada. Con respecto a los resultados sin PCA para los descriptores el rendimiento es levemente menor, sin embargo, la reducción del tamaño del descriptor es importante. Podríamos decir que habiendo reducido la dimensionalidad en un 50 % obtuvimos un rendimiento similar que para el descriptor sin transformar con PCA, para MLP.

Tabla 5.6: Porcentajes de Reconocimiento sobre conjunto de Testeo CENPARMI usando MLP para descriptores basados en TW y PCA.

Descriptor	Dimensión	Nueva Dimensión (PCA)	Retiene % Varianza	MLP (% Reconocidos)
Sin Preprocesar	256			89.05
		128	91.65	90.65
		64	81.81	*91.75*
LL1	64			91.65
		48	95.51	82.10
T2	64			*92.15*
		48	97.09	90.45
LL1T2	128			*92.25*
		92	100	91.55
		64	100	*92.10*
		48	96.38	89.70

A continuación se presentan los resultados experimentales para ambas bases de datos, utilizando SVM: la Tabla 5.8 muestra los resultados para CENPARMI, mientras que la Tabla 5.9 muestra los resultados para MNIST (15000 patrones de entrenamiento). Los porcentajes de reconocimiento mejoran para los casos donde se utiliza PCA, superando también a los experimentos realizados con MLP. Es decir, aplicando las técnicas de TW y PCA se ha logrado reducir la dimensión del descriptor en un 75 y 87.50 %, logrando mejorar el rendimiento en la clasificación.

La Figura 5.7 compara los resultados obtenidos para ambas bases de datos y para los descriptores TW y TW-PCA con clasificadores SVM.

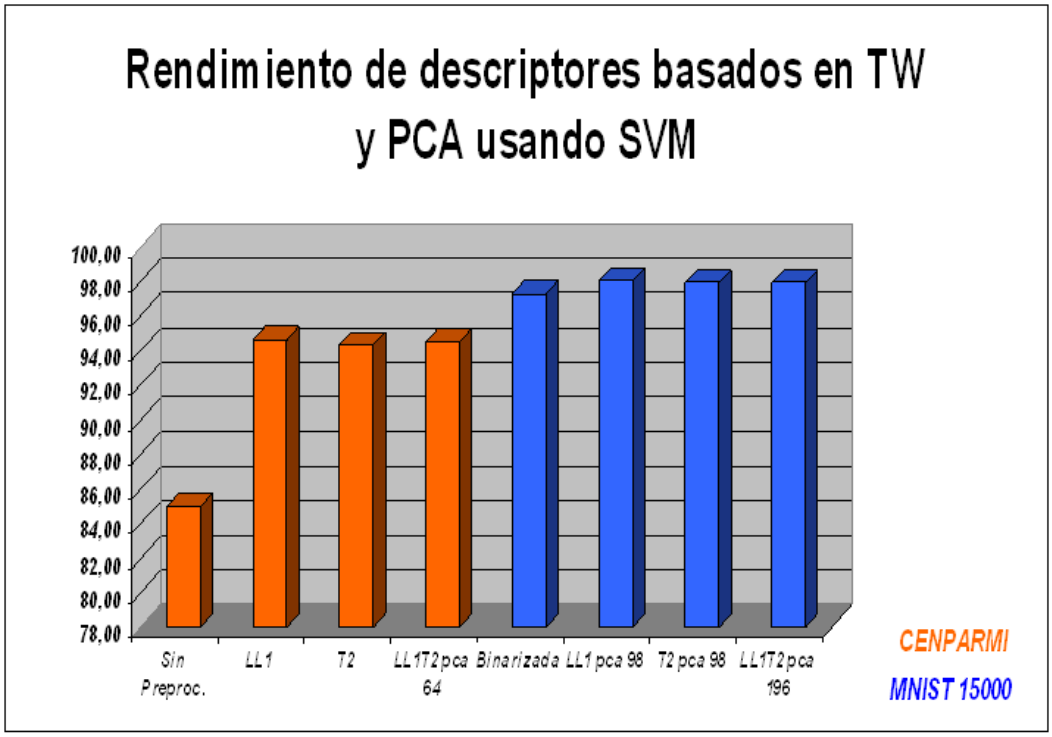


Figura 5.7: Comparación del rendimiento de descriptores basados en TW y PCA luego de clasificar con SVMs para las bases CENPARMI y MNIST asociada a 15000 patrones de entrenamiento.

Tabla 5.7: Porcentajes de Reconocimiento sobre conjunto de Testeo MNIST usando MLP para descriptores basados en TW y PCA, para un conjunto de 15000 patrones de entrenamiento.

Descriptor	Dimensión	Nueva Dimensión (PCA)	Retiene % Varianza	MLP (% Reconocidos)
Binarizada	784			96.22
LL1	196			*97.00*
		98	98.40	96.72
T2	196			96.98
		98	98.28	96.52
LL1T2	392			*97.01*
		196	100	96.70

Tabla 5.8: Porcentajes de Reconocimiento sobre conjunto de Testeo CENPARMI usando SVM para descriptores basados en TW y PCA.

Descriptor	Dimensión	SVM (% Reconocidos)
Sin Preprocesar	256	85.00
LL1	64	94.60
T2	64	94.35
LL1T2	PCA 64	94.50

Tabla 5.9: Porcentajes de Reconocimiento sobre conjunto de Testeo MNIST usando SVM para descriptores basados en TW y PCA, para un conjunto de 15000 patrones de entrenamiento - *: mejores resultados.

Descriptor	Dimensión	SVM (% Reconocidos)	
Binarizada	784	97.33	
LL1	196	97.59	
	PCA 98	98.04	*
T2	196	97.54	
	PCA 98	97.94	*
LL1T2	392	97.60	
	PCA 196	97.96	*

5.3. Conclusiones

En este Capítulo presentamos diferentes descriptores basados en características de multirresolución para representar dígitos manuscritos. Para esto hemos utilizado la Transformada Wavelet CDF 9/7. Estos descriptores permitieron aumentar el rendimiento en la clasificación disminuyendo la dimensionalidad de las imágenes en un 50 y 75 %. La aplicación de PCA sobre algunas de estas características permitió disminuír aún más el tamaño del descriptor (en un 87.50 %), logrando rendimientos en la clasificación similares o mayores a los ya obtenidos. Es importante resaltar que el tipo de preprocesamiento aplicado permitió aumentar la calidad de la representación al mismo tiempo que se logró una alta tasa de reducción del tamaño del descriptor. Finalmente diremos que, los descriptores conformados por la aproximación de primer nivel de la transformada CDF 9/7; el segundo nivel de dicha transformada; y ambos, es decir, aproximación de primer nivel y segundo nivel completo, y la aplicación de PCA sobre éstos, ha permitido representar en forma eficiente a los dígitos en la clasificación, lográndose un rendimiento mayor que la imagen sin preprocesar en todos los casos, y habiendo reducido la dimensión del patrón en un 75 y 87.50 %.

Capítulo 6

Combinación de Clasificadores

El objetivo de este capítulo es presentar el método de combinación de múltiples clasificadores como una estrategia alternativa de diseño de un reconocedor de patrones, que permite mejorar el rendimiento. Se presentan las estrategias clásicas de combinación así como también la Estrategia Bayesiana con detección de patrones Ambiguos (EBA) propuesta en este trabajo, junto con la experimentación asociada al problema del reconocimiento de dígitos manuscritos.

6.1. Introducción

La combinación de múltiples clasificadores se propone como un método que permite mejorar la precisión en el reconocimiento, en comparación con la utilización de clasificadores por separado (simples), es decir sin combinar [26][7]. Rahman [55] presenta un estudio sobre métodos para combinar clasificadores aplicados al reconocimiento de caracteres que incluye el análisis de distintas formas de organización. Por ejemplo, una combinación horizontal o paralela es frecuentemente utilizada cuando se prioriza una alta precisión en el reconocimiento, mientras que una estructura secuencial o en cascada se utiliza cuando es necesario acelerar la clasificación en grandes conjuntos de datos categorizados.

El rendimiento de un sistema reconocedor de patrones que utiliza múltiples clasificadores, no sólo depende de la estrategia de combinación de resultados intermedios seleccionada, sino que depende también de la complementariedad o diversidad de los clasificadores participantes. Esta complementariedad puede surgir de la aplicación de diferentes enfoques como por ejemplo, modificar los conjuntos de entrenamiento o las caracterísiticias extraídas, la estructura del clasificador, los métodos de aprendizaje, entre otras variantes. En los últimos años se desarrollaron métodos para generar múltiples clasificadores a través de la exploración de la muestras del conjunto de entrenamiento en función de una característica

específica, los cuales están recibiendo una creciente atención, como es el caso de las técnicas de Bagging [62] y Boosting [63]. Para el reconocimiento de caracteres, la combinación de clasificadores basada en la utilización de diferentes preprocesamientos o extracción de características, ha demostrado ser efectiva [7].

6.2. Métodos Clásicos

El problema de la combinación de múltiples clasificadores es un tema de investigación actualmente en desarrollo en el área de Reconocimiento de Patrones. En un comienzo, el objetivo principal estaba orientado al diseño de un clasificador con excelente rendimiento y también a la reducción de la dimensionalidad de los descriptores utilizados. Actualmente, la combinación apropiada de varios clasificadores con un buen rendimiento (que no tiene por qué ser el mejor) dedicados al tratamiento de características diferentes y complementarias es una técnica que permite obtener un reconocimiento de alta calidad. Asociados a este enfoque surgen varios problemas o interrogantes, como por ejemplo, determinar la cantidad de clasificadores adecuada para un determinado problema, los tipos de clasificadores apropiados y las características a considerar, entre otros [26].

En general, los métodos que se utilizan para combinar las decisiones de múltiples clasificadores dependen del tipo de información producida por los clasificadores individuales.

Sea P el espacio de datos de entrada consistente en M conjuntos mutuamente excluyentes, $P = C_1 \cup \ldots \cup C_M$, con cada C_i, $\forall i \in \Lambda = \{1, 2, \ldots, M\}$ representando un conjunto de patrones específicos denominado *clase* (por ejemplo, $M = 10$ para el problema de reconocimiento de dígitos). Para una muestra x extraída de P, la tarea del clasificador e consiste en asignar un índice $j \in \Lambda \cup \{M+1\}$ como rótulo que representa que x pertenece a la clase C_j si $j \neq M+1$, y en caso que $j = M+1$ significando que x es un patrón rechazado por e [26]. De esta forma, consideramos al clasificador como una función que recibe una entrada x y proporciona una salida j tal que $e(x) = j$.

De acuerdo a las salidas que pueden producir los clasificadores individuales, se definen tres tipos de problemas para los cuales se aplican distintas técnicas de combinación:

Tipo 1: Nivel abstracto o salida compuesta por una única clase. Dados K clasificadores individuales $e_k, k = 1, \ldots, K$, cada uno de los cuales asigna un rótulo j_k a una entrada dada x, es decir, produce un evento $e_k(x) = j_k$, se utilizan dichos eventos para construir un clasificador integrado E que asigna a la entrada x un rótulo definitivo j tal que $E(x) = j, j \in \Lambda \cup M + 1$.

Tipo 2: Nivel de rangos o listas categorizadas. Dada una entrada x, cada e_k produce un subconjunto $L_k \subseteq \Lambda$ tal que todos los rótulos en L_k forman una lista ordenda por rango. El problema consiste en

utilizar los eventos $e(x) = L_k, K = 1, \ldots, K$ para construir un clasificador E tal que $E(x) = j, j \in \Lambda \cup \{M + 1\}$.

Tipo 3: Nivel de mediciones. Dada una entrada x, cada e_k produce un vector de números reales $M_e(k) = m_k(1), \ldots, m_k(M)^T$, donde $m_k(i)$ representa en qué medida e_k considera que x pertenece a la clase i. El problema consiste en utilizar los eventos $e(x) = M_e(k), k = 1, \ldots, K$, para construir un clasificador E tal que $E(x) = j, j \in \Lambda \cup \{M + 1\}$.

Los tres tipos de problemas descriptos en su conjunto cubren un amplio espectro de aplicaciones. En el Tipo 1, los clasificadores individuales pueden estar basados en diferentes teorías y metodologías (por ejemplo, uno podría ser un clasificador estadístico y otro estar basado en un método sintáctico), ya que lo que interesa es el resultado en el nivel abstracto. Por esta razón se puede afirmar que los problemas de Tipo 1 cubren todas las áreas dentro del Reconocimiento de Patrones, considerándose una de las categorías más útiles debido a su generalidad.

A diferencia de este Tipo más general, los problemas de Tipo 3 requieren que todos los clasificadores individuales provean un resultado en el nivel de mediciones, y que los mismos puedan ser transformados representando el mismo tipo de información tal que su combinación tenga sentido. Un método asociado a este Tipo es el Promedio de Clasificadores Bayesianos, donde todas las salidas de los clasificadores individuales deben representar una probabilidad a posteriori [26]. Por otro lado, los problemas de Tipo 2 requieren que todos los clasificadores den un resultado en el nivel de rango, es decir, produzcan una salida consistente en una lista de clases posibles con un *ranking* asociado. El mismo puede estar implementado como un orden, o valores de confiabilidad, o distancias, entre otras posibilidades también consideradas como salidas en el nivel de mediciones. La combinación de listas categorizadas está especialmente indicado para problemas de reconocimiento de patrones con muchas clases, donde el hecho de que una clase aparezca en los primeros lugares de la lista sea significativo para la clasificación; como ejemplo de aplicación mencionaremos el reconocimiento de palabras [56].

En los últimos años los métodos asociados al Tipo 1 han recibido una creciente atención debido a su simplicidad, robustez y a su alta precisión en los resultados. Estos incluyen el Voto por Mayoría y sus variantes como el Voto por Mayoría Ponderado [54], Formulación Bayesiana [26], Teoría de Dempster-Shafer [56], entre otros, aplicados al problema de OCR.

Describiremos brevemente algunos de estos métodos con el objeto de presentar las bases para el desarrollo de la estrategia de combinación de clasificadores propuesta en la presente tesis.

6.2.1. Voto por Mayoría

El sistema de Voto por Mayoría tiene un gran número de variantes asociadas construidas sobre el mismo principio subyacente. Las estrategias para resolver esta técnica responden a distintos interrogantes

que surgen a la hora de resolver la combinación de los clasificadores individuales, como por ejemplo: la decisión final debería responder únicamente a la cantidad de expertos que votaron a la clase ganadora sin importar la competencia de cada uno de ellos a la hora de clasificar o la decisión final debería responder al voto del experto más competente sin tener en cuenta el consenso de la mayoría?

El enfoque básico del Voto por Mayoría [26] considera la existencia de K expertos independientes, cada uno de los cuales produce una respuesta única, según lo expresado para los métodos asociados al Tipo 1 de problemas o Nivel Abstracto. La decisión final $E(x) = j$ que surge de los eventos $e_k(x) = j_k, k = 1, \ldots, K$, está dada por la clase que consigue más de la mitad de los votos, es decir, cuando al menos q expertos coinciden, siendo q

$$q = \begin{cases} \frac{K}{2} + 1 & \text{si} \quad K \text{ es par} \\ \\ \frac{K+1}{2} & \text{si} \quad K \text{ es impar} \end{cases} \tag{6.1}$$

En este contexto, puede demostrarse que el éxito de la estrategia del Voto por Mayoría depende directamente de la confiabilidad de las respuestas dadas por cada uno de los expertos participantes [54].

Definimos la función $T_k(x \in C_i)$ como,

$$T_k(x \in C_i) = \begin{cases} 1 & \text{cuando} \quad e_k = i, i \in \Lambda \\ \\ 0 & \text{en otro caso} \end{cases} \tag{6.2}$$

la estrategia del Voto por Mayoría puede expresarse según [26] como,

$$E(x) = \begin{cases} j & \text{si} \quad T_E(x \in C_j) = \text{máx}_{i \in \Lambda}\, T_E(x \in C_i) > \frac{K}{2} \\ \\ M + 1 & \text{de otro modo} \end{cases} \tag{6.3}$$

donde,

$$T_E(x \in C_i) = \sum_{k=1}^{K} T_k(x \in C_i),\ i = 1, \ldots, M \tag{6.4}$$

Una expresión más general permite definir un parámetro α, $0 < \alpha \leq 1$ para regular la cantidad de votos necesarios para que una clase sea considerada ganadora, según muestra la fórmula 6.5,

$$E(x) = \begin{cases} j & \text{si} \quad T_E(x \in C_j) = \text{máx}_{i \in \Lambda}\, T_E(x \in C_i) \geq \alpha * K \\ \\ M+1 & \text{de otro modo} \end{cases} \tag{6.5}$$

Notar que la fórmula 6.3 es un caso especial de 6.5 para $\alpha = 0{,}5 + \epsilon$, siendo $\epsilon > 0$ arbitrariamente pequeño.

En algunos casos puede suceder que más de una clase haya recibido una cantidad importante de votos. En estas situaciones tomar como respuesta la clase más votada puede no ser muy preciso, en consecuencia [26] propone la siguiente regla de Voto por Mayoría,

$$E(x) = \begin{cases} j & \text{si} \quad T_E(x \in C_j) = \text{máx}_1,\ \ \text{máx}_1 - \text{máx}_2 \geq \alpha * K \\ \\ M+1 & \text{de otro modo} \end{cases} \tag{6.6}$$

donde,

$$\begin{aligned} \text{máx}_1 &= \text{máx}_{i \in \Lambda}\, T_E(x \in C_i) \\ \\ \text{máx}_2 &= \text{máx}_{i \in \Lambda - \{j\}}\, T_E(x \in C_i) \end{aligned} \tag{6.7}$$

lo que implica que para que una clase sea considerada como la respuesta final, debe tener asociada una cantidad de votos lo suficientemente grande [26].

6.2.2. Voto por Mayoría Ponderado

Un simple refinamiento a la estrategia de Voto por Mayoría consiste en considerar la confiabilidad de las respuestas de cada uno de los clasificadores individuales multiplicando cada salida por un peso [55]. Los pesos w_k que expresan la competencia comparativa entre los expertos participantes, se definen como una lista de fracciones tal que

$$\sum_{k=1}^{K} w_k = 1 \tag{6.8}$$

siendo K el total de expertos. Cuanto mayor es la competencia de un experto, mayor es el valor del w asociado. De esta forma, denotamos la decisión de un experto e_k que asocia a una entrada x con la clase

i^{th}, como d_{ik}, para $i = 1, \ldots, M$, siendo M el total de clases. La decisión que surge de la combinación de la salida de los distintos clasificadores para la clase i, d_i^{com}, se define como:

$$d_i^{com} = \sum_{k=1}^{K} w_k * d_{ik} \tag{6.9}$$

La decisión final d^{com} estará dada por:

$$d^{com} = \max_{i=1,\ldots,M} d_i^{com} \tag{6.10}$$

Otras variantes sobre la forma de determinar los pesos han sido presentadas para este método, como por ejemplo el Voto por Mayoría Ponderado que utiliza un índice de confianza asociado con cada clase, o el Voto por Mayoría Restringido asociado a una lista de valores de confianza que expresan la competencia comparativa de los clasificadores [54].

6.2.3. Regla de Combinación Bayesiana

La Regla de Combinación Bayesiana [26] permite considerar el rendimiento de cada experto sobre las muestras de entrenamiento para cada clase. En particular, la matriz de confusión R_k de cada clasificador k sobre el conjunto de entrenamiento se utiliza como indicador del rendimiento de cada experto. Para un problema con M clases más la opción de rechazo, R_k es una matriz de dimensión M x $(M + 1)$, donde cada entrada $\eta_{ij}^{(k)}$ denota el número de patrones rotulados con la clase i y que fueron asignados a la clase j por el clasificador, para $j \leq M$. La opción $j = M + 1$ representa la cantidad de patrones que fueron rechazados.

De la matriz R_k podemos obtener el número total de muestras $N^{(k)}$ en el conjunto utilizado como,

$$N^{(k)} = \sum_{i=1}^{M} \sum_{j=1}^{M+1} \eta_{ij}^{(k)} \tag{6.11}$$

y el número total de patrones que pertenecen a la clase i, $\eta_{i.}^{(k)}$, sumando sobre la fila i según indica la fórmula 6.12,

$$\eta_{i.}^{(k)} = \sum_{j=1}^{M+1} \eta_{ij}^{(k)}, i = 1, \ldots, M \tag{6.12}$$

Obtenemos también el número de muestras que fueron asignadas a la clase j por un experto k, $\eta_{.j}^{(k)}$, sumando sobre la columna j,

$$\eta_{.j}^{(k)} = \sum_{i=1}^{M} \eta_{ij}^{(k)}, j = 1, \ldots, M+1 \tag{6.13}$$

Frente a la presencia de K expertos, tendremos K matrices R_k, $1 \leq k \leq K$. En consecuencia, la probabilidad condicional de que un patrón x pertenezca a la clase i, dado que el experto k lo asoció con la clase j se estima como,

$$P(x \in C_i \,|e_k(x) = j) = \frac{\eta_{ij}^{(k)}}{\eta_{.j}^{(k)}} = \frac{\eta_{ij}^{(k)}}{\sum_{i=1}^{M} \eta_{ij}^{(k)}}, i = 1, \ldots, M \tag{6.14}$$

El valor obtenido en la ecuación 6.14 representa la precisión del experto k cuando asigna la clase i a una entrada x.

Dado un patrón x tal que los resultados de la clasificación de cada uno de los K expertos es $e_k(x) = j_k$ para $1 \leq k \leq K$, se define un valor de confianza sobre la afirmación de que la entrada x pertenece a la clase i como,

$$\text{bel}(i) = P(x \in C_i \,|e_1(x) = j_1, \ldots, e_K(x) = j_K) \tag{6.15}$$

Aplicando la fórmula de Bayes, y asumiendo que las decisiones de los expertos son independientes, el valor $bel(i)$ puede estimarse como,

$$\text{bel}(i) \approx \frac{\prod_{k=1}^{K} P(x \in C_i \,|e_k(x) = j_k)}{\sum_{i=1}^{M} \prod_{k=1}^{K} P(x \in C_i \,|e_k(x) = j_k)}, 1 \leq i \leq M. \tag{6.16}$$

Para cada patrón de entrada x, el mismo es asignado a la clase j si $bel(j) > bel(i)$ para todo $i \neq j$ y $bel(j) > \alpha$ para un umbral α definido. En caso de no cumplirse con estas condiciones el patrón es rechazado. También se rechaza en caso que $e_k(x) = M+1$ para todo k, es decir, todos los clasificadores coincidieron en rechazar la entrada x.

Los resultados obtenidos con este método dependen del valor del umbral α elegido. A medida que α crece, aumenta el grado de certeza esperado en la decisión final. De esta forma el error decrecería, pero el porcentaje de patrones correctamente clasificados podría también ser bajo [56].

Para más referencias sobre combinación bayesiana ver [26] [96] [56].

6.3. Estrategia Bayesiana con detección de Patrones Ambiguos (EBA)

Debido a que nuestra propuesta de reconocedor está orientada a una aplicación general más allá del reconocimiento de dígitos manuscritos, la estrategia presentada está orientada a problemas de Tipo 1 o de nivel abstracto, donde la salida de los clasificadores individuales está compuesta por una única clase.

Como ya hemos mencionado en el Capítulo 2, el clasificador presentado en esta tesis consta de dos niveles: un primer nivel de clasificadores individuales y un segundo nivel consistente en un módulo analizador que combina las respuestas dadas en el primer nivel para resolver una respuesta final. Vimos también que la respuesta final no era solamente decir a qué clase pertenecería el patrón ingresado, sino también indicar si el mismo era considerado ambiguo o no y en el primer caso indicar con qué otra clase podría confundirse. El sistema propuesto, en principio, no rechaza patrones aunque su diseño permite implementar fácilmente estrategias para estos casos.

Describiremos aquí el segundo nivel del sistema, es decir el módulo analizador, y luego presentaremos la experimentación pertinente.

Los elementos que utiliza el módulo analizador para combinar las salidas de los clasificadores individuales son: una tabla de confiabilidad cuyos valores expresan cuán confiable es la respuesta dada por cada uno de los clasificadores individuales; y dos parámetros que denominamos *umbral de confiabilidad* y *distancia mínima* relacionados con la detección de patrones ambiguos.

Construcción de la Tabla de Confiabilidad

Cada clasificador del primer nivel está asociado a: un determinado descriptor que surge de características extraídas de los patrones de entrada, un método de clasificación determinado y a cierta arquitectura. La construcción de la tabla de confiabilidad se basa en un enfoque probabilístico bayesiano para expresar cuán confiable es la respuesta dada por cada clasificador, y en función de esto poder determinar la respuesta final.

Utilizando la definición de probabilidad condicional y la regla de multiplicación, definimos la probabilidad $P(\{x \in C\}/\{e_k(x) = C\})$ tal que un patrón de entrada x pertenezca a la clase C dado que el clasificador individual e_k respondió C para la entrada x, como

$$P(\{x \in C\}/\{e_k(x) = C\}) = \frac{P(\{e_k(x) = C\}/\{x \in C\})P(\{x \in C\})}{P(\{e_k(x) = C\})} \tag{6.17}$$

donde,

$P(\{x \in C\})$ es la probabilidad de que el patrón de entrada x pertenezca a la clase C, estimada utilizando el conjunto de datos rotulados de entrenamiento;
$P(\{e_k(x) = C\})$ se estima utilizando el clasificador ya entrenado e_k. Se asume que la respuesta C dada por cada clasificador individual e_k, dado un patrón de entrada x es independiente de las respuestas de los otros clasificadores individuales;
$P(\{e_k(x) = C\}/\{x \in C\})$ se estima a partir de las salidas correctas del clasificador e_k, para los patrones de entrada pertenecientes a la clase C.
Cada elemento de la tabla de confiabilidad corresponde a un valor que representa la probabilidad *a posteriori* calculada en base a la fórmula 6.17 para un clasificador y una clase determinada, utilizando el conjunto de entrenamiento y los clasificadores ya entrenados. Es decir, cuando un clasificador individual dice que el patrón de entrada x pertenece a la clase C, cuán confiable es esta respuesta en función de su rendimiento con el conjunto de entrenamiento? Si para las respuestas C ese clasificador no fue muy preciso, entonces tendrá un valor bajo asociado en la tabla.

Estrategia de clasificación

En la etapa de clasificación, un patrón de entrada x es ingresado al sistema y como salida del primer nivel se obtiene una nueva representación de x, a través de las respuestas o votos de los clasificadores individuales.

Ya como parte de las funciones del módulo analizador, se calcula un puntaje asociado con cada clase votada s_C basado en los valores de la Tabla de Confiabilidad que evalúan el rendimiento de cada clasificador. Una primera aproximación para la resolución de este cálculo se presenta en la fórmula 6.18,

$$s_C = \sum_{e_k \in E_C} r_{C,e_k} \tag{6.18}$$

donde, C indica la clase seleccionada para la cual se está calculando el puntaje, e_k indica el clasificador individual que se está considerando, E_C es el conjunto de los clasificadores individuales que votaron a la clase C, r_{C,e_k} valor extraído de la Tabla de Confiabilidad para la clase C y el clasificador e_k.

De esta forma, para cada clase votada se calcula el puntaje asociado sumando los valores de confiabilidad extraídos de la Tabla, tal que una clase con mayor puntaje, en principio, implicaría respuestas más confiables y mayor cantidad de votos para esa clase.

Como hemos visto, una de las principales dificultades a la hora de clasificar, y en particular cuando hablamos de escritura manuscrita, es el tratamiento de valores extremos o *outliers* y de *patrones ambiguos*, dado que las distorsiones que presentan hacen difícil su correcta clasificación. Estos patrones tienen

la particularidad de estar alejados del valor medio de la clase con la que fueron rotulados, y por esta razón podrían ser incorrectamente asociados con otra clase a la que estuvieran más cercanos.

Para considerar este tipo de situaciones de forma tal de poder aumentar el poder discriminativo del clasificador, consideramos la distancia del patrón (representado por su vector de características) al valor medio de la clase asignada por el clasificador individual: si el descriptor está cercano al valor medio, se asume que el patrón está bien definido y pertenece a dicha clase; un patrón alejado de la media podría ser considerado candidato a *patrón ambiguo* en la salida final del sistema. Esta información es utilizada como un factor de refuerzo de los valores extraídos de la Tabla de Confiabilidad, como muestra la fórmula 6.19 que indica finalmente cómo se calcula el puntaje por cada clase votada s_C.

$$s_C = \sum_{e_k \in E_C} r_{C,e_k} \frac{1}{d(x_f, \mu_{e_k,C})} \tag{6.19}$$

donde, C indica la clase seleccionada para la cual se está calculando el puntaje, e_k indica el clasificador individual que se está considerando, E_C es el conjunto de los clasificadores individuales que votaron a la clase C, r_{C,e_k} valor extraído de la Tabla de Confiabilidad para la clase C y el clasificador e_k,x_{e_k} patrón representado por su vector de características asociado al clasificador e_k (tener en cuenta que cada clasificador individual puede estar asociado a características diferentes),$d(x_{e_k}, \mu_{e_k,C})$ distancia normalizada entre el vector de características del patrón de entrada x y el valor medio de la clase C para el conjunto de entrenamiento considerado para el clasificador e_k. Notar que, en realidad, la representación del patrón y el cálculo de la media por clase se realizan considerando la(s) característica(s) extraída(s) de los datos de entrada a la(s) cual(es) está dedicada dicho clasificador.

En la fórmula 6.19, cada valor de probabilidad extraído de la Tabla de Confiabilidad se divide por la distancia normalizada entre el patrón de entrada y la media de la clase asignada, de forma tal que patrones cercanos a la media incrementan el puntaje total, mientras que los patrones más alejados disminuyen el valor de confiabilidad considerado.

Definición de la salida del sistema

Para definir la salida del reconocedor es necesario fijar los valores de dos parámetros: el umbral de confiabilidad y la distancia mínima. El *umbral de confiabilidad* permite decidir si un patrón será ambiguo o no para el sistema. Frente al caso de un patrón ambiguo, la *distancia mínima* permite considerar la o las clases con la que dicho patrón podría confundirse.

Una vez calculados los puntajes por clase votada según la fórmula 6.19, el módulo analizador identifica a la clase con mayor puntaje asociado, y lo compara con el umbral de confiabilidad. Si el puntaje de la clase ganadora supera el umbral, entonces el sistema considera que el patrón está bien definido y

que pertenece a la clase ganadora. En caso que el puntaje máximo no supere el umbral de confiabilidad, entonces se considera que el patrón es ambiguo, es decir no está claramente definido. En este caso, el puntaje máximo es comparado con el puntaje más cercano de los calculados. Si ambos están lo suficientemente cerca, es decir, su distancia no supera al parámetro de distancia mínima, entonces la salida del sistema indicará que el patrón ingresado es ambiguo y que además podría pertenecer a alguna de las clases asociadas con los dos puntajes máximos.
Esta lógica podría ejemplificarse considerando los siguientes casos:

1. Todos los clasificadores individuales votan a la misma clase: aqui se calcula un único puntaje asociado con la clase votada por todos. Dicho puntaje deberá superar el umbral de confiabilidad, y el patrón se considerará bien definido. El caso donde todos menos uno votaron a la misma clase podría ser similar, y serviría para ejemplificar el caso de robustez frente a fallos (para una respuesta correcta): al tener varios clasificadores, si alguno se equivoca no afecta la salida final del sistema.

2. Una clase fue votada por mayoría: habrá que considerar además de la cantidad de votos, también la calidad de los mismos junto con el grado de distorsión del patrón ingresado, medido con la distancia entre el patrón y la media de la clase votada. Todos estos elementos elevan la calidad de la clasificación, por ejemplo, sobre la estrategia de Voto por Mayoría.

3. Ninguna clase tiene mayoría de votos: en este caso, lo más probable es que no haya un puntaje máximo muy alejado del resto, y que dicho máximo no supere el umbral de confiabilidad. Si los clasificadores votaron diferentes clases, podría decirse que el patrón no está muy definido y podría confundirse con otras clases.

Los valores de los parámetros *umbral de confiabilidad* y *distancia mínima* se determinan experimentalmente sobre la base de información provista por el conjunto de datos de entrenamiento completo en la etapa de ajuste del clasificador, teniendo en cuenta los valores máximo y mínimo de los puntajes por clase y las distancias entre los mismos. La variación de dichos parámetros permite ajustar la salida del sistema sin necesidad de reentrenar los clasificadores individuales. En la Sección 6.4.1 se analizan distintos ejemplos.

Notar que en este contexto, la salida del sistema consiste en indicar si el patrón es ambiguo o no, y la o las clases a las que podría pertenecer. Además, el hecho de considerar clasificadores individuales asociados a distintas características extraídas de los datos de entrada en el primer nivel del sistema, y todos los elementos utilizados en el modulo analizador, permiten de alguna manera explicar o justificar la salida del reconocedor.

A continuación presentaremos la experimentación que nos permitió corroborar la eficiencia y conveniencia de este enfoque.

6.4. Experimentación

Los resultados experimentales que se presentan para este Capítulo se basan en la utilización de los sistemas reconocedores que muestra la Figura 6.1, orientados a probar las distintas técnicas de combinación. Cada clasificador está dedicado a una característica direccional extraída con las Máscaras de Kirsch (ver Capítulo 4), y un quinto clasificador se dedica a la característica global o patrón completo. Uno de los sistemas reconocedores utiliza SVM para clasificar. Éste fue aplicado sobre las bases CENPARMI y MNIST, con un preprocesamiento adicional consistente en aplicar la Transformada Wavelet CDF 9/7 sobre las cinco características, para utilizar luego la subbanda de aproximación de primer nivel LL1 binarizada, generando un descriptor de tamaño un cuarto con respecto al tamaño original (ver Capítulo 5). El otro sistema reconocedor utiliza clasificadores SOM, y fue aplicado a la base CENPARMI.

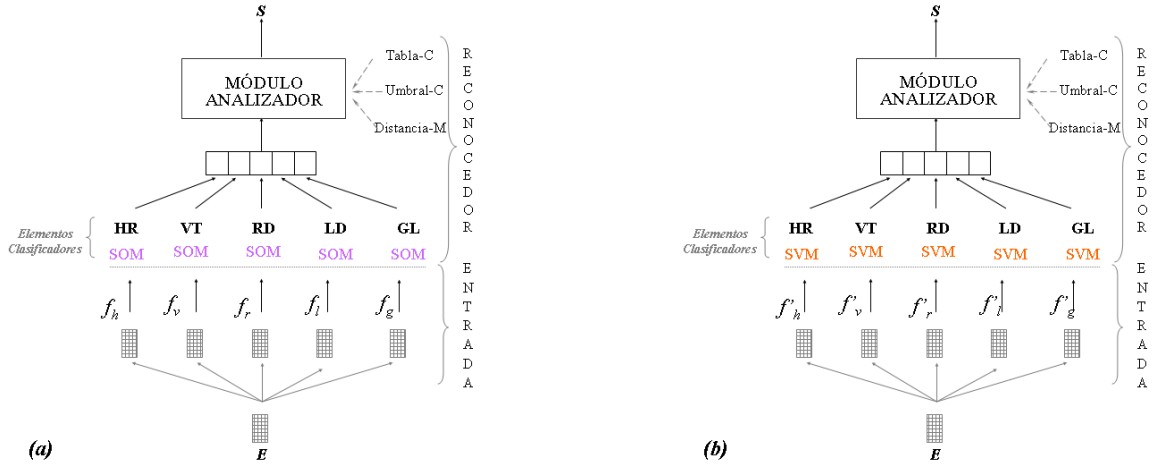


Figura 6.1: Sistemas reconocedores que combinan (a) SOMs asociados a características direccionales, (b) SVM asociados a características direccionales procesadas con la Transformada Wavelet CDF 9/7.

Presentamos, en primer lugar, resultados relacionados con la implementación de la EBA, para luego compararlos con la aplicación de las estrategias clásicas descriptas en este Capítulo. Es decir, las estrategias aplicadas para la combinación de los cinco clasificadores son: 1) Voto por Mayoría, 2) Voto por Mayoría Ponderado, 3) Regla de Combinación Bayesiana, 4) EBA - Estrategia Bayesiana propuesta.

6.4.1. EBA

La Tabla 6.1 muestra los porcentajes de reconocimiento para los clasificadores individuales SOM construidos sobre la base CENPARMI [49] donde las imágenes normalizadas a 16x16 píxeles generaron un descriptor de dimensión 256.

Tabla 6.1: Rendimiento de cada clasificador SOM asociado a una caracteristica de preprocesamiento diferente, sobre el conjunto de testeo para la base CENPARMI - (Características: GL global o patrón completo, HR horizontal, VT vertical, RD diagonal derecha, LD diagonal izquierda).

SOM asociado a característica	% Patrones reconocidos	SOM asociado a característica	% Patrones reconocidos
HR	84.60	VT	82.05
RD	83.45	LD	86.25
GL	88.90		

Tabla 6.2: Tabla de Confiabilidad - SOMs asociados a características direccionales; HR - horizontal, VT- vertical, RD - diagonal derecha, LD - diagonal izquierda, GL - característica global - valores para CENPARMI.

Clase	HT	VT	RD	LD	GL
0	0.925	0.938	0.932	0.958	0.964
1	0.955	0.929	0.964	0.957	0.973
2	0.903	0.931	0.936	0.920	0.963
3	0.850	0.868	0.857	0.887	0.926
4	0.972	0.916	0.970	0.941	0.968
5	0.926	0.898	0.856	0.943	0.933
6	0.967	0.970	0.970	0.968	0.983
7	0.933	0.851	0.955	0.912	0.943
8	0.919	0.966	0.905	0.934	0.964
9	0.919	0.849	0.915	0.914	0.964

Para la aplicación de la EBA se construyó la Tabla de Confiabilidad según la fórmula 6.17. Los valores obtenidos pueden observarse en la Tabla 6.2. Los valores de los parámetros Umbral de Confiabilidad

(UC) y Distancia Mínima (DM) fueron fijados experimentalmente. La Tabla 6.3 muestra resultados del reconocimiento para distintos valores de estos parámetros.

Tabla 6.3: Resultados del reconocimiento (%) para el sistema con SOMs asociados a características direccionales para CENPARMI - UC: Umbral de Confiabilidad - DM: distancia mínima - *: mejores resultados.

	UC	DM	Correctas (incluye ambiguos)	Correctas (única respuesta)	Error
	2.0	0.5	91.00	90.50	9.00
*	6.0	3.0	*94.50*	80.60	5.50
	6.0	2.0	94.20	84.20	5.80
*	9.0	3.0	94.50	80.60	5.50
	15.0	1.5	93.65	86.20	6.35

Analizando los resultados del reconocimiento, observamos que para UC $2{,}0$ y DM $0{,}5$ prácticamente no se detectan patrones ambiguos; esto se debe a que el valor del umbral es bajo en comparación a los puntajes máximos obtenidos para las clases votadas. A medida que el valor del umbral de confiabilidad aumenta, los patrones asociados a puntajes altos para la clase ganadora serán considerados bien definidos, mientras que el resto será considerado como patrones con cierto grado de similitud con elementos de otras clases. Como ya mencionamos en la sección anterior, la utilización de la Distancia Mínima posibilita introducir una segunda clase en la salida para estos patrones, siempre y cuando el puntaje asociado esté lo suficientemente cercano al puntaje de la clase ganadora. En la segunda y tercera fila de la Tabla 6.3 se observa que para el mismo valor de UC la DM decrece, y como consecuencia el número de patrones asociado a una única clase se ha incrementado.

La Tabla 6.4 muestra algunos resultados del proceso de reconocimiento correspondientes a los patrones del conjunto de testeo que se observan en la Figura 6.2. Los dígitos de la primer columna de la figura están bien definidos para el sistema. De hecho, casi todos los elementos clasificadores votaron a la misma clase para estos patrones, como puede observarse en la primera fila de cada clase en la Tabla 6.4. Sin embargo, sabemos que en la definición de la salida no sólo se toma en cuenta la cantidad de votos por clase. La segunda y tercer columna de la Figura 6.2 muestran patrones que fueron considerados ambiguos por el sistema. En la Tabla 6.4 puede observarse que la salida indica dos clases posibles para estos dígitos, coincidiendo una de ellas con el rótulo del patrón. Los votos aparecen distribuídos entre diferentes clases, por lo tanto, el puntaje asociado con las clases ganadoras es menor que el correspondiente a un patrón bien definido. Un análisis visual de la Figura 6.2 revela que el segundo '0' es, en realidad, similar a un '6', y el tercer '0' tiene ruido. Para el resto de los patrones ambiguos, las formas

Tabla 6.4: Algunos resultados del reconocimiento correspondientes a los patrones del conjunto de testeo CENPARMI, para estrategia EBA con UC $= 6{,}0$ y DM $= 3{,}0$. Indica patrones ambiguos y votos por cada SOM para las características direccionales HR - horizontal, VT- vertical, RD - diagonal derecha, LD - diagonal izquierda y GL - global.

Clase	Salida Sistema	Ambig	Voto HR	VT	RD	LD	GL
0	0	No	0	0	0	0	0
0	0 ó 6	Sí	0	0	6	0	6
0	0 ó 2	Sí	0	5	2	0	1
2	2	No	2	2	2	2	2
2	2 ó 6	Sí	2	6	2	2	2
2	0 ó 2	Sí	0	5	0	2	2
5	5	No	3	5	5	5	5
5	3 ó 5	Sí	3	3	3	5	5
5	6 ó 5	Sí	5	6	5	6	6

que presentan pueden asociarse con más de una clase. Por ejemplo, el segundo patrón rotulado como ‘5’ bien podría ser un ‘3’ incompleto, mientras que el tercer ‘5’ podría asociarse con un ‘6’ incompleto.

Otro sistema reconocedor utilizado y compuesto por SVMs, también fue implementado sobre la base CENPARMI. En este caso, la aplicación de la TW sobre las características direccionales y la característica global, disminuyó la dimensión del descriptor de 256 a 64 (lo que representa un 75 % de reducción). La Tabla 6.5 muestra los porcentajes de reconocimiento para los clasificadores individuales SVM, mientras que las tablas 6.6 y 6.7 muestran los valores de confiabilidad y los resultados para distintos valores de parámetros UC y DM, respectivamente.

Comparando los resultados para CENPARMI correspondientes al sistema compuesto por SOMs y al compuesto por SVMs, vemos que éste último mejora el porcentaje de patrones correctamente clasificados de $94{,}50$ a $95{,}45$, a su vez que disminuye el error, es decir la cantidad de patrones mal clasificados. Además, el preprocesamiento utilizado en este sistema permitió reducir la dimensión del descriptor en un 75 %, lo cual es una gran ventaja a la hora de reducir costos de entrenamiento y clasificación.

Para la totalidad de los patrones MNIST se construyó un sistema basado en SVM con el preprocesamiento consistente en la extracción de características direccionales usando Máscaras de Kirsch y la aplicación de la TW CDF 9/7, como se describió previamente. Este preprocesamiento permitió obtener un descriptor de dimensión 196 (784 era el tamaño original), efectuando una reducción de un 75 %, como

Figura 6.2: Patrones del conjunto de testeo CENPARMI correctamente clasificados. Filas asociadas con ejemplos de las clases ‘0’, ‘2’ y ‘5’.

Tabla 6.5: Rendimiento de cada clasificador SVM asociado a una caracteristica de preprocesamiento diferente, sobre el conjunto de testeo para la base CENPARMI - (Características: GL global o patrón completo, HR horizontal, VT vertical, RD diagonal derecha, LD diagonal izquierda).

SVM asociado a característica	% Patrones reconocidos	SVM asociado a característica	% Patrones reconocidos
HR	81.05	VT	80.40
RD	82.95	LD	86.75
GL	90.60		

en el caso CENPARMI. La Tabla 6.8 muestra los porcentajes de reconocimiento para los clasificadores individuales SVM.

La Tabla 6.9 muestra los valores de confiabilidad utilizados en el módulo analizador para definir la salida del sistema. La Tabla 6.10 muestra los resultados para distintos valores de parámetros UC y DM [95].

El porcentaje de patrones reconocidos obtenido con este sistema es bastante alto (*99.11 %*), inclusive en comparación con trabajos publicados, como se verá en el Capítulo 7. El análisis sobre la variación de los valores de parámetros UC y DM y los resultados en el reconocimiento sobre la Tabla 6.10 es similar al realizado para los otros sistemas. A medida que el UC aumenta, los patrones bien definidos serán aquellos asociados a una clase ganadora con puntaje alto, probablemente votada por varios elementos clasificadores. A su vez, el resto de los patrones será considerado como dígitos con cierto grado de similitud con otras clases, pudiendo generar una salida de más de una clase. Un UC bajo podría incrementar el

Tabla 6.6: Tabla de Confiabilidad - SVMs asociados a características direccionales con TW; HR - horizontal, VT- vertical, RD - diagonal derecha, LD - diagonal izquierda, GL - característica global - valores para CENPARMI.

Clase	HT	VT	RD	LD	GL
0	0.872	0.946	0.921	0.925	1.000
1	0.951	0.900	0.745	0.930	0.930
2	0.872	0.860	0.971	0.760	0.974
3	0.725	0.882	0.868	0.912	0.946
4	0.952	0.851	0.860	0.900	0.907
5	0.786	0.733	0.857	0.921	0.884
6	0.927	0.841	0.902	0.952	0.976
7	0.949	0.903	0.976	0.895	0.929
8	0.875	0.778	0.714	0.806	0.944
9	0.892	0.791	0.838	0.795	0.892

porcenteaje de patrones mal clasificados ya que un patrón dudoso (los elementos clasificadores votaron distintas clases) podría ser clasificado con una sola clase, que quizás no fuera la correcta. Como ejemplo, la Tabla 6.11 muestra algunos resultados del proceso de reconocimiento sobre patrones del conjunto de testeo que se observan en la Figura 6.10.

En la primer columna de la Figura 6.3 se observan patrones que están bien definidos; de hecho, todos los elementos clasificadores votaron a la misma clase (ver Tabla 6.11). No ocurrió lo mismo con los patrones de la segunda y tercer columna, considerados ambiguos para el sistema. Por ejemplo, cualquiera podría afirmar que el tercer '2' es similar a un '7', o que el tercer patrón rotulado como '5' podría ser un '8' incompleto.

6.4.2. Estrategias clásicas de combinación

La estrategia de Voto por Mayoría ha sido implementada según la fórmula 6.6 y con valores de $\alpha = 0{,}1$ y $K = 5$, este último representando la cantidad de clasificadores considerados. En este contexto, una clase con puntaje máximo a un voto de diferencia de la segunda clase más votada, será considerada la salida del sistema. Por otro lado, los patrones rechazados son aquellos que han obtenido la misma cantidad máxima de votos para dos clases. Por ejemplo, de los cinco clasificadores podría haber sucedido que dos votaran a una clase A, otros dos a otra clase B, y que otro clasificador asociara el patrón de

Tabla 6.7: Resultados del reconocimiento (%) para el sistema con SVMs asociados a características direccionales para CENPARMI - UC: Umbral de Confiabilidad - DM: distancia mínima - *: mejor resultado

	UC	DM	Correctas (incluye ambiguos)	Correctas (única respuesta)	Error
	4.0	2.5	95.35	81.85	4.65
	4.0	3.0	95.40	79.95	4.60
*	5.0	3.0	*95.45*	77.90	4.55
	6.0	3.0	95.45	77.85	4.55

Tabla 6.8: Rendimiento de cada clasificador SVM asociado a una caracteristica de preprocesamiento diferente, sobre el conjunto de testeo para la base completa MNIST - (Características: GL global o patrón completo, HR horizontal, VT vertical, RD diagonal derecha, LD diagonal izquierda).

SVM asociado a característica	% Patrones reconocidos	SVM asociado a característica	% Patrones reconocidos
HR	93.40	VT	95.52
RD	95.22	LD	95.82
GL	97.33		

entrada con una clase C; de esta forma habría empate entre las clases A y B. Otro caso sería que todos los clasificadores votaran clases diferentes.

En el caso de la estrategia de Voto por Mayoría Ponderado, hemos definido los pesos asociados a cada reconocedor según su rendimiento utilizando patrones del conjunto de entrenamiento, de forma tal de cumplir con lo planteado en la fórmula 6.8. En esta estrategia consideramos la clase ganadora como la asociada al mayor puntaje calculado según 6.10, y en caso de haber empate entre dos o más clases, el patrón es rechazado.

La aplicación de la Regla de Combinación Bayesiana descripta en la Subsección 6.2.3, se implementó en base a los valores de la matriz de confusión generada por el conjunto de entrenamiento, y estimando los valores de confianza según la fórmula 6.16 para las clases votadas. La salida del sistema corresponde a la clase con mayor valor de confianza.

Las Tablas 6.12, 6.13 y 6.14 muestran los resultados de aplicar las estrategias de combinación sobre los tres sistemas reconocedores descriptos en la subsección 6.4.1. La EBA ha dado los porcentajes de

Tabla 6.9: Tabla de Confiabilidad - SVMs asociados a características direccionales con TW; HR - horizontal, VT- vertical, RD - diagonal derecha, LD - diagonal izquierda, GL - característica global - valores para MNIST.

Clase	HT	VT	RD	LD	GL
0	0.925	0.978	0.967	0.975	0.992
1	0.959	0.972	0.974	0.983	0.983
2	0.957	0.965	0.971	0.961	0.972
3	0.905	0.952	0.960	0.944	0.967
4	0.970	0.960	0.960	0.955	0.958
5	0.938	0.966	0.937	0.979	0.978
6	0.965	0.985	0.960	0.967	0.980
7	0.958	0.932	0.972	0.957	0.977
8	0.866	0.945	0.920	0.939	0.973
9	0.927	0.923	0.958	0.928	0.950

reconocimiento más altos en todos los casos, permitiendo detectar y clasificar los patrones ambiguos, algo que no realizan las otras estrategias. Además, en esta estrategia todos los patrones son asociados con alguna clase, mientras que en las otras un porcentaje del conjunto de testeo es rechazado.

La Figura 6.4 presenta un gráfico comparativo de los resultados de las distintas estrategias para los sistemas construidos para CENPARMI y MNIST.

Tabla 6.10: Resultados del reconocimiento (%) para el sistema con SVMs asociados a características direccionales para MNIST - UC: Umbral de Confiabilidad - DM: distancia mínima - *: mejor resultado

	UC	DM	Correcta (incluye ambiguos)	Correcta (única respuesta)	Error
	4.0	3.0	98.65	94.94	1.35
	6.0	3.0	98.97	93.68	1.03
	6.0	4.0	99.08	90.14	0.92
*	6.0	5.0	*99.11*	89.55	0.89

Tabla 6.11: Resultados sobre el conjunto de testeo para los dígitos de la Figura 6.3

Clase	Respuesta Sistema	Ambig.	voto HR	voto VT	voto RD	voto LD	voto GL
2	2	No	2	2	2	2	2
2	2 u 8	Sí	3	2	8	2	2
2	7 ó 2	Sí	2	7	7	9	2
3	3	No	3	3	3	3	3
3	3 u 8	Sí	8	3	3	3	5
3	3 u 8	Sí	8	3	8	3	3
5	5	No	5	5	5	5	5
5	5 ó 0	Sí	5	2	5	0	0
5	5 u 8	Sí	3	8	5	5	8

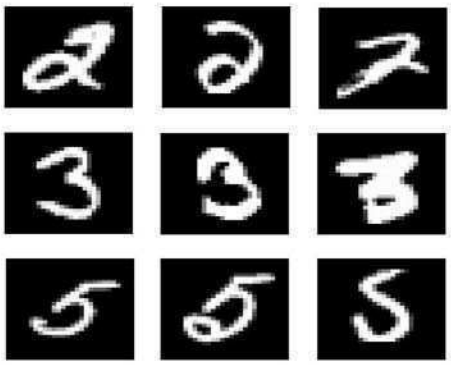


Figura 6.3: Patrones de testeo correctamente clasificados para el sistema basado en SVM para MNIST. En cada fila: clases ‘2’, ‘3’ y ‘5’ respectivamente.

Tabla 6.12: Estrategias de Combinación de clasificadores para el Sistema Reconocedor basado en SOMs y en características direccionales para la base CENPARMI.

	Estrategia	% Patrones reconocidos	% Patrones rechazados	% Patrones mal clasif.	Detecta Pat. ambiguos
1)	Voto por Mayoría	90.00	3.10	6.90	No
2)	Voto por Mayoría Ponderado	91.15	0.15	8.70	No
3)	Regla de Comb. Bayesiana	90.95	1.30	7.75	No
4)	*Estrategia Bayesiana Propuesta*	*94.50*	–	*5.50*	*Sí*

Tabla 6.13: Estrategias de Combinación de clasificadores para el Sistema Reconocedor basado en SVMs y en características direccionales y CDF 9/7 para la base CENPARMI.

	Estrategia	% Patrones reconocidos	% Patrones rechazados	% Patrones mal clasif.	Detecta Pat. ambiguos
1)	Voto por Mayoría	91.35	3.95	4.70	No
2)	Voto por Mayoría Ponderado	92.25	1.60	6.15	No
3)	Regla de Comb. Bayesiana	76.40	17.90	5.70	No
4)	*Estrategia Bayesiana Propuesta*	*95.45*	–	*4.55*	*Sí*

Tabla 6.14: Estrategias de Combinación de clasificadores para el Sistema Reconocedor basado en SVMs y en características direccionales y CDF 9/7 para la base MNIST.

	Estrategia	% Patrones reconocidos	% Patrones rechazados	% Patrones mal clasif.	Detecta Pat. ambiguos
1)	Voto por Mayoría	97.73	0.72	1.55	No
2)	Voto por Mayoría Ponderado	97.91	0.31	1.78	No
3)	Regla de Comb. Bayesiana	97.46	0.38	2.16	No
4)	*Estrategia Bayesiana Propuesta*	*99.11*	–	*0.89*	*Sí*

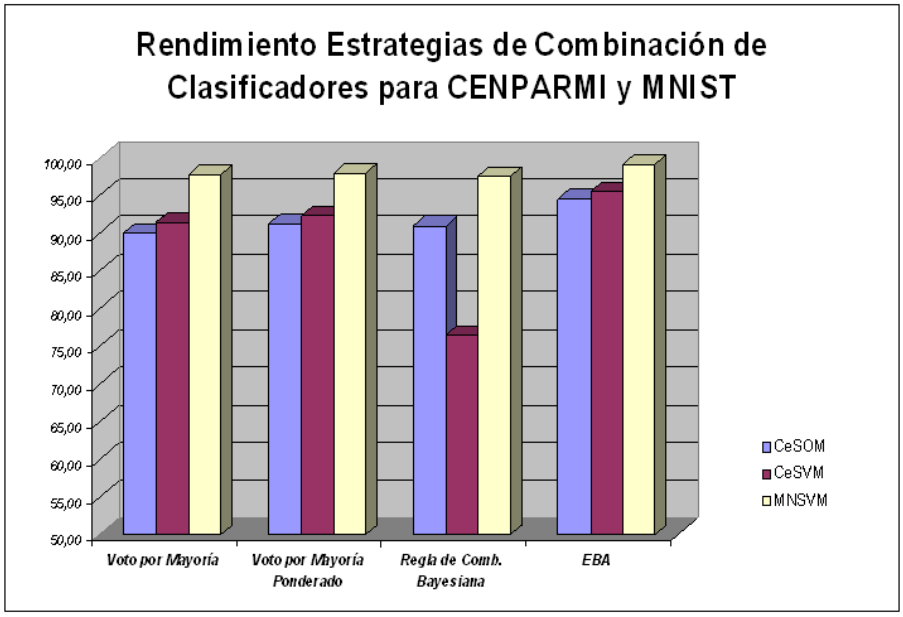


Figura 6.4: Estrategias de Combinación de clasificadores para CENPARMI y MNIST.

6.5. Conclusiones

En este Capítulo se ha presentado la estrategia de combinación Bayesiana que detecta patrones ambiguos *EBA*, para la combinación de múltiples clasificadores. También se ha descripto el problema general relacionado con la combinación y las estrategias clásicas que constituyen una alternativa de diseño para mejorar el rendimiento.

El análisis se centró en los métodos asociados a los problemas de Tipo 1, reconocidos por su simplicidad, robustez y alta precisión. Básicamente, dada una cierta cantidad de respuestas de clasificadores o votos a combinar, la estrategia de Voto por Mayoría toma en cuenta la cantidad de votos por clase para definir una salida final. La estrategia de Voto por Mayoría Ponderado asigna un peso a cada clasificador en función de su rendimiento con el conjunto de entrenamiento, de forma tal de mejorar la precisión de las respuestas finales. Por otro lado, la Regla de Combinación Bayesiana utiliza la fórmula de Bayes y el rendimiento de cada clasificador por clase, para estimar un valor de confianza por cada clase votada en función del comportamiento de todo el sistema. Este valor de confianza permite definir la salida del sistema. En las tres estrategias mencionadas, los patrones que no pueden ser asociados con alguna clase son rechazados. La Estrategia Bayesiana EBA utiliza la fórmula de Bayes y el rendimiento de cada clasificador por clase para estimar la probabilidad de éxito de cada clasificador individual por cada respuesta dada. En función de este valor de confiabilidad, de una medida de distancia del patrón de entrada a la media de la clase votada, y de la utilización de dos parámetros, se define la salida del sistema. Un aspecto que nos interesa destacar a nivel de análisis de la salida, es que la estrategia presentada detecta los patrones ambiguos, indica con qué clases podrían confundirse, y no rechaza patrones de entrada (aunque permite incorporar esta característica), es decir, siempre asocia la entrada con la clase más cercana, indicando si el patrón es considerado ambiguo o no. Esta es una diferencia con las otras estrategias que indican el rechazo de patrones y no distinguen entre patrones bien definidos y los dudosos. Los resultados experimentales muestran que la EBA supera a las otras estrategias en cuanto a porcentaje de patrones reconocidos para distintos tipos de clasificadores individuales utilizados, disminuyendo el error y aumentando la calidad de la respuesta.

Capítulo 7

Sistema Clasificador con Tratamiento de Ambigüedad - SCLAM

El objetivo de este capítulo es presentar la experimentación asociada a la construcción de los sistemas reconocedores finales SCLAM, incluyendo la selección del preprocesamiento, presentando estrategias para la selección de los elementos clasificadores, y comparando los resultados finales con resultados de la literatura considerados representativos.

7.1. Características direccionales con Máscaras de Kirsch y TW-PCA

En el Capítulo 4 hemos presentado distintas técnicas para extracción de características sobre las imágenes de dígitos manuscritos, y en el Capítulo 5 propusimos varios descriptores basados en el uso de la Transformada Wavelet (TW) y Análisis de Componentes Principales (PCA), con muy buen rendimiento para el problema tratado. En esta Sección presentamos resultados sobre características direccionales y TW-PCA con el objeto de obtener un conjunto de clasificadores que permita construir los sistemas reconocedores finales.

De esta forma, trabajamos con las cuatro características direccionales extraídas de los dígitos y sobre ellas aplicamos los descriptores basados en la TW y PCA (TW-PCA) presentados en el Capítulo 5. El entrenamiento de diferentes clasificadores utilizando SVM (método con el que hemos obtenido los mejores resultados) nos dió un conjunto de clasificadores a partir del cual elegimos los mejores elementos para construir el sistema reconocedor.

En las Tablas 7.1 y 7.2 se presentan los resultados sobre ambas bases y para los clasificadores entrenados.

Tabla 7.1: Porcentajes de Reconocimiento sobre conjunto de Testeo *CENPARMI* usando SVM para descriptores basados en características direccionales, TW y PCA.

Descriptor	Dimensión	HR	VT	% RD	LD	GL
LL1	64	90.10	91.35	91.60	93.00	94.60
T2	64	89.70	91.25	91.10	92.55	94.35
LL1T2	PCA 64	89.80	91.40	90.95	92.65	94.50

Tabla 7.2: Porcentajes de Reconocimiento sobre conjunto de Testeo *MNIST* usando SVM para descriptores basados en características direccionales, TW y PCA, para un conjunto de *15000* patrones de entrenamiento.

Descriptor	Dimensión	HR	VT	% RD	LD	GL
LL1	PCA 98	96.38	96.53	96.26	96.38	98.04
T2	PCA 98	96.20	96.46	95.98	96.26	97.94
LL1T2	PCA 196	96.29	96.62	95.98	96.29	97.96

Los resultados para MNIST utilizando el conjunto de entrenamiento de 15000 patrones permitió comparar los porcentajes obtenidos con resultados previos presentados en este trabajo, asi como también evaluar el rendimiento de los descriptores con un conjunto de datos compuesto por un cuarto de las imágenes del conjunto de entrenamiento original de la base. Teniendo en cuenta que el rendimiento con los tres descriptores considerados (aplicados sobre la características direccionales y global) ha sido muy bueno y similar, se decidió utilizar los mismos para la base completa.

La Tabla 7.3 muestra los resultados para MNIST completa.

En términos generales, se observa que el rendimiento es muy bueno y similar para las distintas características direccionales, aunque el mayor porcentaje de patrones reconocidos se obtiene con la característica global. La utilización de la base de datos MNIST completa ha permitido mejorar los resultados sobre el conjunto de testeo, en comparación con las pruebas realizadas con MNIST con 15000 patrones en el conjunto de entrenamiento.

A partir de ahora utilizaremos las bases CENPARMI y MNIST COMPLETA para la experimentación orientada a construir los sistemas reconocedores finales.

Tabla 7.3: Porcentajes de Reconocimiento sobre conjunto de Testeo *MNIST* usando SVM para descriptores basados en características direccionales, TW y PCA, para la base *COMPLETA*.

Descriptor	Dimensión	HR	VT	% RD	LD	GL
LL1	PCA 98	97.35	97.67	97.65	97.68	98.64
T2	PCA 98	97.27	97.60	97.65	97.68	98.54
LL1T2	PCA 196	97.30	97.66	97.61	97.70	98.59

7.2. Selección de Elementos Clasificadores

Como ya hemos mencionado en el Capítulo 6, la selección de clasificadores para su combinación no es una tarea trivial. Uno de los criterios aplicados, por su buen desempeño para el problema del reconocimiento de dígitos manuscritos, es la utilización de diferentes características cada una asociada con un clasificador independiente y extraídas del mismo conjunto de datos original. En base a esta idea presentamos otras estrategias de selección de subconjuntos de clasificadores, su combinación mediante la técnica EBA descripta en la Sección 6.3, y los resultados obtenidos.

E1) Una primera idea ya utilizada en varios trabajos ([95] [49] [30]), consiste en combinar las cuatro características direccionales y la global. En este trabajo se agrega el preprocesamiento usando la Transformada Wavelet y PCA, indicado en la sección anterior, sobre las características direccionales y sobre la característica global. De esta forma, tendremos un sistema reconocedor por cada descriptor (cada fila de las tablas 7.1 y 7.3) y para cada base de datos. Luego del ajuste de los parámetros *umbral de confiabilidad* (UC) y *distancia mínima* (DM) seleccionamos los mejores resultados, teniendo en cuenta la cantidad de patrones ambiguos detectados y el total de patrones asociados con una única clase, además del porcentaje de reconocimiento total.

La Tabla 7.4 presenta los resultados para la base de datos CENPARMI. En la parte (a) se prioriza los porcentajes totales de patrones correctamente clasificados, mientras que en la (b) se sigue buscando un buen reconocimiento pero se disminuye la cantidad de patrones ambiguos mientras que aumenta el porcentaje de patrones asociados con una única clase.

La Tabla 7.5 presenta los resultados asociados a la base de datos MNIST.

La aplicación de la estrategia *E1* con la técnica de combinación EBA, ha permitido obtener altos porcentajes de reconocimiento para los distintos preprocesamientos utilizados. Para CENPARMI, el porcentaje de reconocimiento más alto fue obtenido con el descriptor *LL1T2 PCA 64* con un *97.40 %* de patrones correctamente clasificados, mientras que para MNIST el porcentaje más alto fue obtenido con

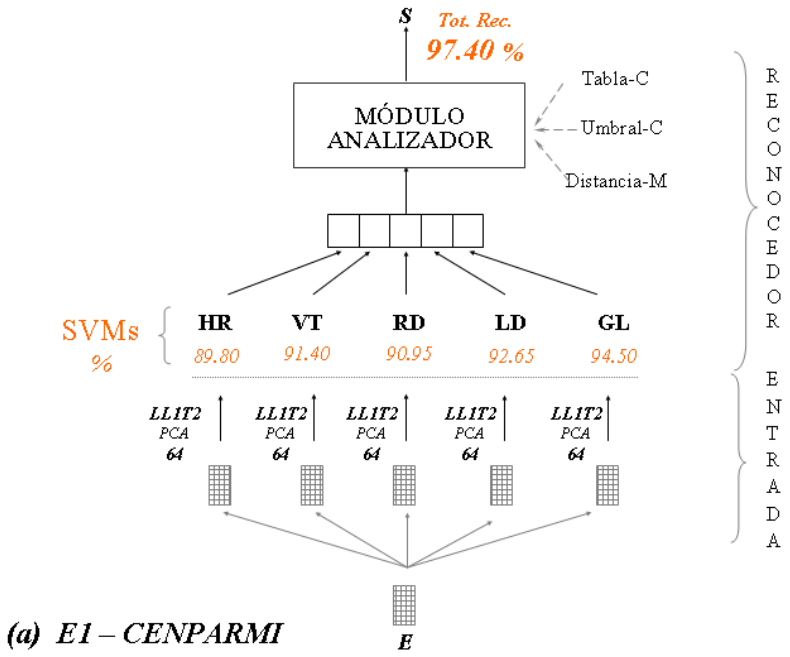
Tabla 7.4: Porcentajes de Reconocimiento sobre conjunto de Testeo *CENPARMI* para la estrategia de combinación E1 - (a) y (b): distintos valores para los parámetros Umbral de Confiabilidad (UC) y Distancia Mínima (DM) propios de la estrategia de combinación EBA.

Descriptor	UC	DM	% Total (incluye ambiguos)	% Reconocidos (única clase)	% Ambig.	% Error
LL1	9.0	7.0	97.35	81.45	15.90	2.65
T2	8.5	8.5	97.25	80.65	16.60	2.75
LL1T2 PCA 64	9.0	6.5	*97.40* *(a)*	81.00	16.40	2.60
LL1	6.5	3.0	96.85	90.30	6.55	3.15
T2	8.5	3.0	96.60	90.30	6.30	3.40
LL1T2 PCA 64	9.0	3.0	*97.05* *(b)*	90.50	6.55	2.95

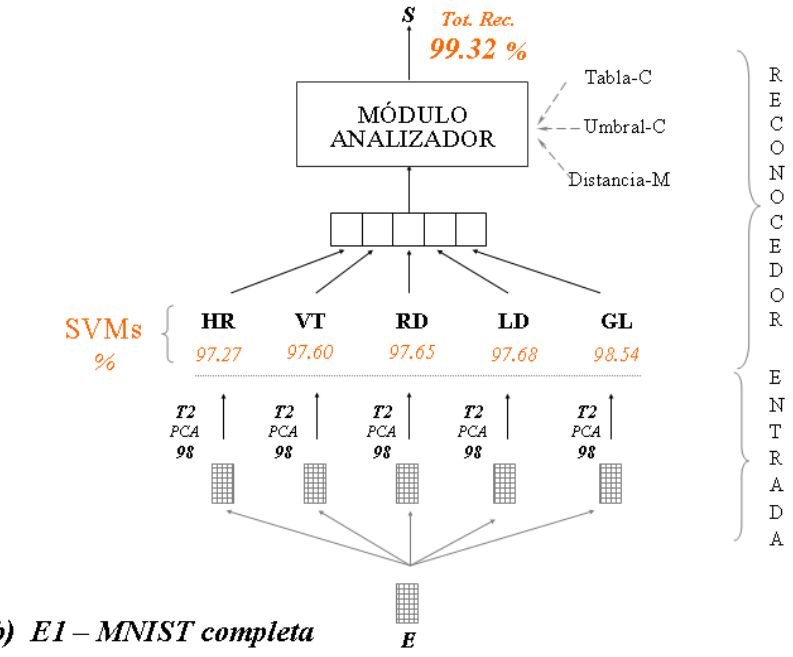
Tabla 7.5: Porcentajes de Reconocimiento sobre conjunto de Testeo *MNIST* para la estrategia de combinación E1, utilizando la base de datos completa. UC: Umbral de Confiabilidad; DM: Distancia Mínima, parámetros propios de la estrategia de combinación EBA.

Descriptor	UC	DM	% Total (incluye ambiguos)	% Reconocidos (única clase)	% Ambig.	% Error
LL1 PCA 98	8.0	8.0	99.29	94.98	4.31	0.71
T2 PCA 98	11.0	8.0	*99.32*	94.74	4.58	0.68
LL1T2 PCA 196	11.0	11.0	99.27	94.89	4.38	0.73

el descriptor *T2 PCA 98* para el sistema con clasificadores entrenados con el conjunto de datos completo, obteniéndose un porcentaje de reconocimiento de *99.32 %*. La Figura 7.1 muestra la estructura del reconocedor con porcentajes asociados, para los mejores resultados.



(a) E1 – CENPARMI



(b) E1 – MNIST completa

Figura 7.1: Sistemas reconocedores asociados a la estrategia de combinación E1 que presentaron los mejores resultados para el conjunto de testeo. (a) CENPARMI y (b) MNIST completa. En cada caso se indica el preprocesamiento, los porcentajes de reconocimiento de cada clasificador individual y el total del sistema.

E2) La estrategia *E2* consiste en utilizar el rendimiento general de cada clasificador presentado en las Tablas 7.1 y 7.3. De esta forma podríamos seleccionar por cada característica direccional y global, qué descriptor fue el más eficiente. Para CENPARMI quedan seleccionados: LL1 para HR, RD, LD y GL, y LL1T2 PCA 64 para VT. Para MNIST quedan LL1 PCA 98 para HR, VT, RD y GL, y LL1T2 PCA 196 para LD.
La Figura 7.2 presenta los resultados para cada conjunto de datos, luego del ajuste de parámetros UC y DM. Se puede observar el rendimiento por cada clasificador individual y el que resulta de la estrategia EBA de combinación de clasificadores.

E3) En la estrategia *E3* consideramos el rendimiento de cada clasificador por característica y por clase, extraído de las Tablas de Confiabilidad asociadas con cada sistema reconocedor (ver Apéndice B). Según lo descripto en el Capítulo 6, la Tabla de Confiabilidad se utiliza para definir la salida del sistema, con todas las propiedades ya mencionadas. Vamos aquí a asignarle una utilidad adicional que es la de permitir seleccionar clasificadores en función del rendimiento por clase. Es decir, incorporar un elemento clasificador al conjunto que sea muy bueno para algunas clases podría ser un aporte importante en el contexto de nuestra estrategia de combinación.
De esta forma, la estrategia *E3* consiste en seleccionar el clasificador que haya tenido mejor rendimiento por clase (según Tablas de Confiabilidad), y en caso de empate elegir el que haya tenido mejor rendimiento general sobre todas las clases teniendo en cuenta la característica y el descriptor (según Tablas 7.1 y 7.3).
Como ejemplo, observamos que para la clase '0' para MNIST, el mejor valor de confiabilidad está asociado con el descriptor GL T2 PCA 98 (ver Tablas B.1 (d), (e) y (f)). En cambio para la clase '1' hay empate entre las características GL asociadas a los descriptores LL1 PCA 98, T2 PCA 98 y LL1T2 PCA 196. Como el mejor rendimiento general sobre todas las clases fue obtenido por el descriptor GL LL1 PCA 98 (ver Tabla 7.3), éste queda seleccionado para la clase del '1'.
Luego de analizar todas las clases, para MNIST completa quedan seleccionados cuatro clasificadores: LL1 PCA 98 para HR, VT y GL, y T2 PCA 98 para GL.
Para CENPARMI quedan seleccionados cinco clasificadores: LL1 para RD, LD y GL, LL1T2 PCA 64 para VT, y T2 para GL.

La Figura 7.3 presenta los mejores resultados obtenidos para la estrategia *E3*.

Los resultados de las estrategias utilizadas en esta Sección serán comparados en la Sección siguiente.

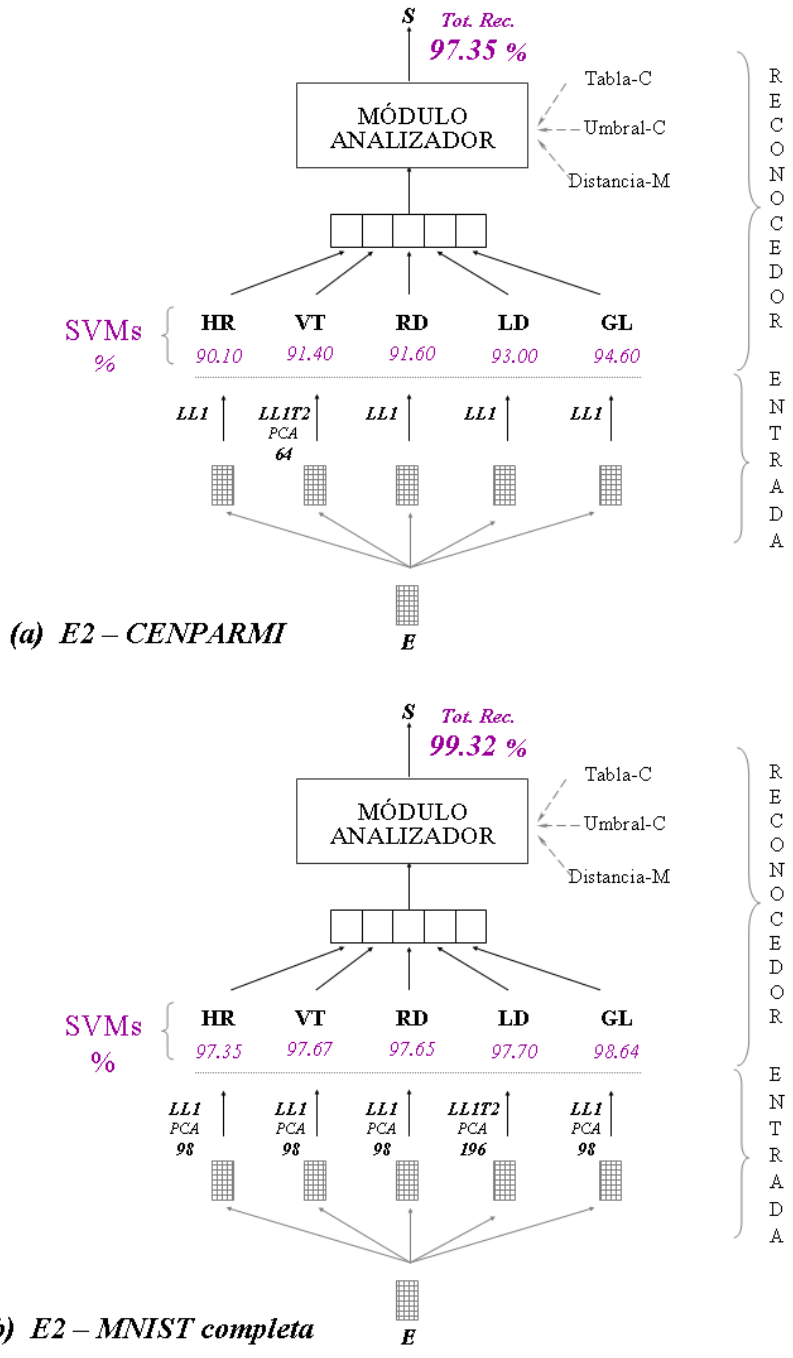
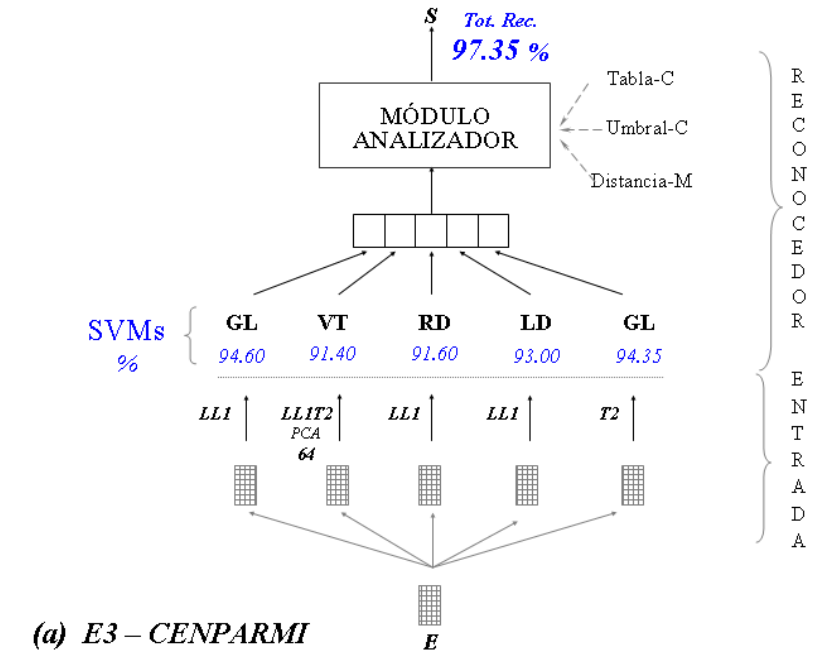
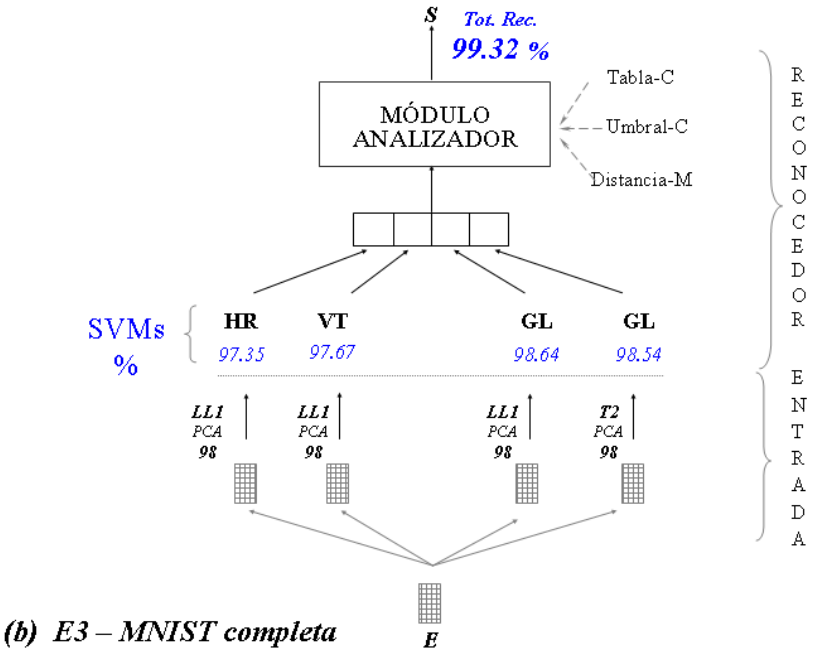


Figura 7.2: Sistemas reconocedores asociados a la estrategia de combinación E2 que presentaron los mejores resultados para el conjunto de testeo. (a) CENPARMI y (b) MNIST completa. En cada caso se indica el preprocesamiento, los porcentajes de reconocimiento de cada clasificador individual y el total del sistema.



(a) E3 – CENPARMI



(b) E3 – MNIST completa

Figura 7.3: Sistemas reconocedores asociados a la estrategia de combinación E3 que presentaron los mejores resultados para el conjunto de testeo. (a) CENPARMI y (b) MNIST completa. En cada caso se indica el preprocesamiento, los porcentajes de reconocimiento de cada clasificador individual y el total del sistema.

7.3. Sistemas de Reconocimiento de Patrones SCLAM

En esta Sección comparamos los resultados obtenidos en la Sección 7.2, luego de haber aplicado los criterios de selección de clasificadores para los distintos conjuntos de datos, con el objeto de elegir los sistemas reconocedores con mejor rendimiento y, así, presentar los resultados finales.

Las Tablas 7.6 y 7.7 comparan los resultados de las tres estrategias presentadas indicando, además, la figura que muestra la estructura de cada sistema reconocedor.

Tabla 7.6: Porcentajes de Reconocimiento sobre conjunto de Testeo *CENPARMI* para clasificadores individuales SVMs y distintas estrategias de combinación.

Sist. Reconocedor	Estrategia	UC	DM	% Total (incluye ambig.)	% Reconocidos (única clase)	% Ambig.	% Error
Fig.7.1 (a)	E1	9.0	6.5	*97.40*	81.00	16.40	2.60
Fig.7.2 (a)	E2	9.0	7.0	97.35	81.45	15.90	2.65
Fig.7.3 (a)	E3	12.0	6.0	97.35	84.05	13.30	2.65

Tabla 7.7: Porcentajes de Reconocimiento sobre conjunto de Testeo *MNIST* COMPLETA para clasificadores individuales SVMs y distintas estrategias de combinación.

Sist. Reconocedor	Estrategia	UC	DM	% Total (incluye ambig.)	% Reconocidos (única clase)	% Ambig.	% Error
Fig.7.1 (b)	E1	11.0	8.0	99.32	94.74	4.58	0.68
Fig.7.2 (b)	E2	11.0	10.0	99.32	94.94	4.38	0.68
Fig.7.3 (b)	E3	6.0	5.0	*99.32*	95.88	3.44	0.68

En el caso CENPARMI, observamos que el mejor resultado fue obtenido con la estrategia *E1*, utilizando todas las características direccionales y la global, mientras que para MNIST la estrategia que mejor ha funcionado fue la *E3*. Notar que para el caso MNIST, aunque los porcentajes totales de reconocimiento coinciden para los tres enfoques, el de la estrategia *E3* presenta un porcentaje mayor de patrones asociados con una única clase, lo cual es deseable.

De esta forma, quedan seleccionados los reconocedores de las Figuras 7.1 (a) y 7.3 (b) como los sistemas finales que han presentado mejor rendimiento (ver Figura 7.4). Notar que para CENPARMI, la dimensión del descriptor ha sido reducida en un 75 % con respecto a las imágenes iniciales para

todos los clasificadores incluidos, mientras que para MNIST la reducción fue del 87.50 % para todos los clasificadores utilizados. Además, en este último caso, la utilización de la estrategia *E3* de selección de clasificadores permitió la utilización de cuatro elementos reconocedores, cantidad menor que la utilizada en los otros sistemas, logrando obtener el rendimiento más alto.

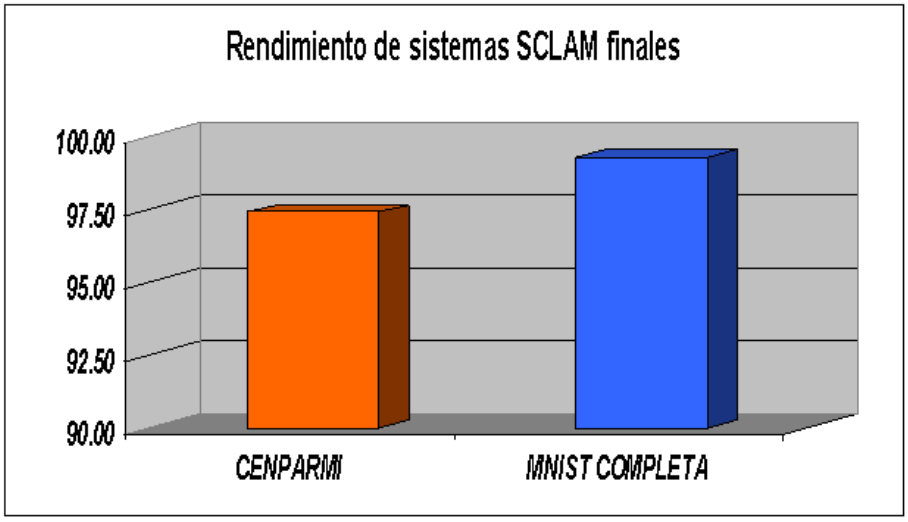


Figura 7.4: Porcentajes de reconocimiento para los sistemas finales SCLAM asociados a los conjuntos de datos CENPARMI (Figura 7.1 (a)) y MNIST completa (Figura 7.3 (b)).

A continuación presentamos ejemplos de las salidas de los reconocedores finales asociados con las bases CENPARMI y MNIST.

La Figura 7.5 presenta dígitos que han sido correctamente clasificados y mal clasificados para el sistema final asociado a CENPARMI, mientras que la Tabla 7.8 muestra los votos recibidos por cada clase por cada patrón de ejemplo. Los patrones de la primera fila de la figura están bien definidos para el sistema, dado que todos los clasificadores votaron a la misma clase. Para los patrones ambiguos los votos se repartieron entre varias clases. Visualmente se observan algunos ejemplos donde la ambigüedad es notoria, como en el caso de la imagen 264 que podría ser un ‘6’ no tan convencional y está rotulado como un ‘1’. Los patrones mal clasificados se presentan incompletos o con formas extrañas que, en algunos casos, confunden a la hora de clasificar, como la imagen 1668 que podría pensarse como un ‘0’ y está rotulada como ‘8’, o la 1637 también rotulada como un ‘8’ pero que parece un ‘4’.

En [46] se evalúa el resultado de clasificar la base CENPARMI con un clasificador de tipo VSVM (SVM virtual). Las imágenes 919, 1663 y 1351, bien definidas para el reconocedor SCLAM, y las número 812 y 1994, consideradas ambiguas, no han sido correctamente clasificadas con el método VSVM, según lo presentado en el trabajo citado. En cuanto a los dígitos mal clasificados, las imágenes 1637 y 1668 no han sido bien clasificadas por ninguno de los dos sistemas.

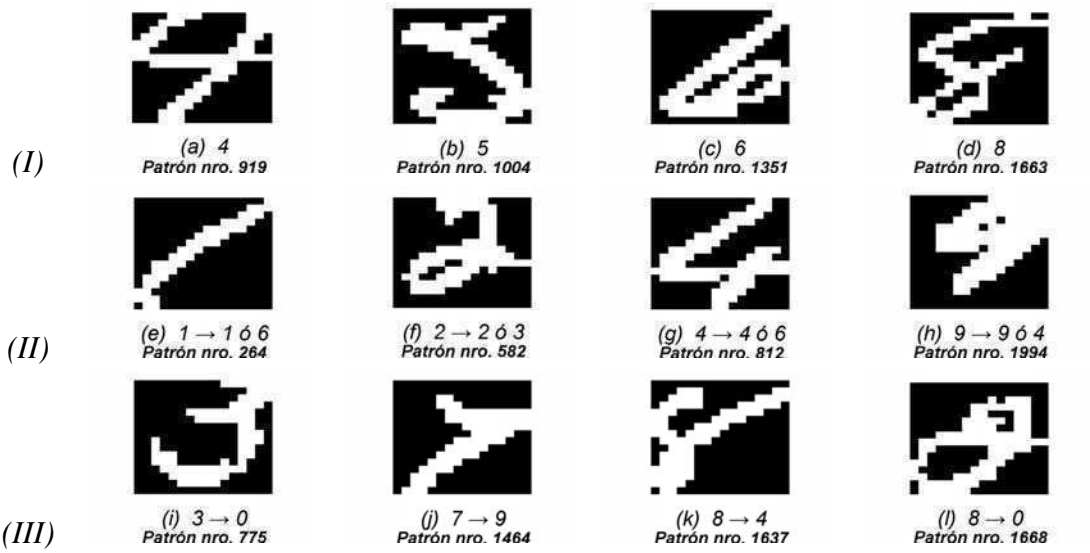


Figura 7.5: Patrones del conjunto de testeo, clasificados por el sistema final de la Figura 7.1 (a) para la base CENPARMI. *(I)*: patrones considerados bien definidos; *(II)*: patrones considerados ambiguos; *(III)*: patrones mal clasificados. En cada caso se indica el rótulo del dígito, y en el caso de ambiguos y mal clasificados, el rótulo y luego la(s) clase(s) asignadas por el sistema.

Tabla 7.8: Algunos resultados del reconocimiento correspondientes al sistema final CENPARMI para los dígitos de la Figura 7.5. Se indican patrones ambiguos y votos por cada SVM para las características HR - horizontal, VT- vertical, GL - global y el descriptor TW-PCA extraído.

	Clase	Salida Sistema	Ambig	Voto HR LL1T2 pca64	VT LL1T2 pca64	RD LL1T2 pca64	LD LL1T2 pca64	GL LL1T2 pca64
(a)	4	4	No	4	4	4	4	4
(b)	5	5	No	5	5	5	5	5
(c)	6	6	No	6	6	6	6	6
(d)	8	8	No	8	8	8	8	8
(e)	1	1 ó 6	Sí	1	6	1	6	1
(f)	2	2 ó 3	Sí	0	2	3	2	8
(g)	4	4 ó 6	Sí	3	4	4	6	6
(h)	9	9 ó 4	Sí	9	9	4	9	9
(i)	3	0	-	0	0	0	0	0
(j)	7	9	-	9	9	2	9	9
(k)	8	4	-	4	4	4	4	4
(l)	8	0	-	0	2	0	0	5

Para la base MNIST completa, la Figura 7.6 presenta los dígitos de ejemplo. La Tabla 7.9 muestra algunos resultados del proceso de reconocimiento con el sistema final, correspondientes a los patrones de dicha figura. Observamos que para los patrones bien definidos, todos los votos fueron para la misma clase. Visualmente algunos son muy claros, como el primer '2', y otros tienen discontinuidades o formas no convencionales, sin embargo fueron correctamente clasificados. Los patrones considerados ambiguos presentan formas poco clásicas para los rótulos asignados, debido a un estilo particular de escritura, a cuestiones de segmentación de las imágenes o características del trazo. Para estos patrones los votos están repartidos en más de una clase, y la salida fue definida en base al puntaje por clase calculado con la estrategia *EBA*. Entre los patrones mal clasificados observamos casos donde imágenes incompletas hacen que el dígito parezca ser de una clase distinta a su rótulo, como el caso de la imagen número 446 que parece un '0' y es un '6'. Otro caso es el de la imagen número 1902 rotulada como '9', pero que un humano podría asociar con un '4', como lo hizo el sistema reconocedor.

En [46] Suen y Tan presentan resultados de cinco clasificadores representativos, como LeNet5 y VSVM (SVM virtuales), y realizan un reporte de los errores cometidos por cada uno, sobre la base MNIST. De los dígitos bien clasificados de la Figura 7.6, las imágenes número 1879, 2036 y 3763 no han podido ser correctamente reconocidas por la mayoría de los clasificadores en el trabajo citado. De los considerados ambiguos, las imágenes número 1233 y 2940 no han podido ser bien clasificadas por ninguno de los clasificadores en [46]. Entre los mal clasificados, las imágenes 1902 y 8409 son errores cometidos por todos los clasificadores, y visualmente estos números tienen formas muy parecidas a elementos de una clase diferente a la que llevan como rótulo.

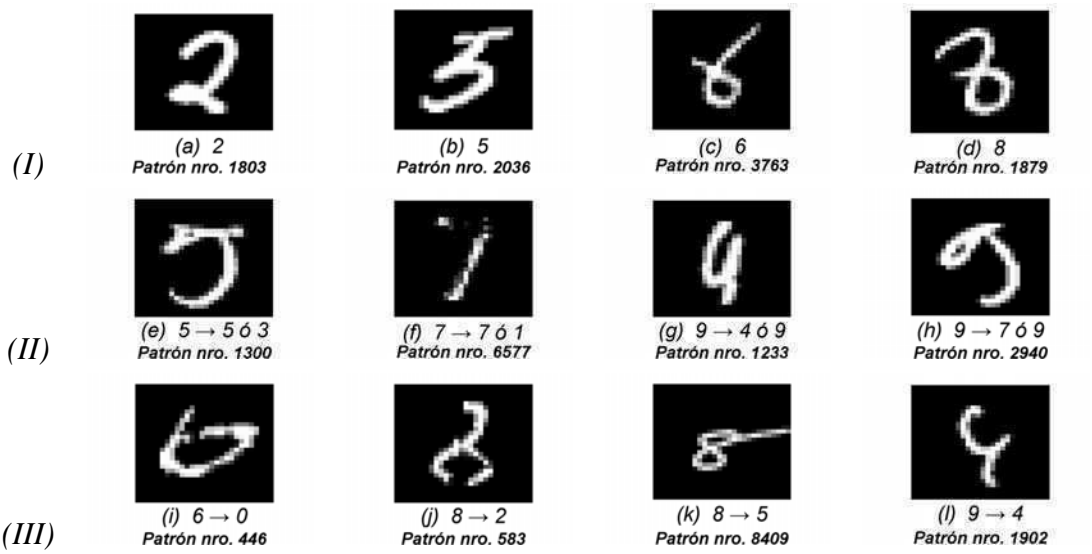


Figura 7.6: Patrones del conjunto de testeo, clasificados por el sistema final de la Figura 7.3 (b) para la base MNIST completa. *(I)*: patrones considerados bien definidos; *(II)*: patrones considerados ambiguos; *(III)*: patrones mal clasificados. En cada caso se indica el rótulo del dígito, y en el caso de ambiguos y mal clasificados, el rótulo y luego la(s) clase(s) asignadas por el sistema.

Tabla 7.9: Algunos resultados del reconocimiento correspondientes al sistema final MNIST completa para los dígitos de la Figura 7.6. Se indican patrones ambiguos y votos por cada SVM para las características HR - horizontal, VT- vertical, GL - global y el descriptor TW-PCA extraído.

	Clase	Salida Sistema	Ambig	Voto HR LL1 pca98	VT LL1 pca98	GL LL1 pca98	GL T2 pca98
(a)	2	2	No	2	2	2	2
(b)	5	5	No	5	5	5	5
(c)	6	6	No	6	6	6	6
(d)	8	8	No	8	8	8	8
(e)	5	5 ó 3	Sí	0	3	5	5
(f)	7	7 ó 1	Sí	1	7	7	7
(g)	9	4 ó 9	Sí	4	9	4	4
(h)	9	7 ó 9	Sí	7	9	7	5
(i)	5	6	-	2	6	3	5
(j)	6	0	-	0	0	0	0
(k)	8	2	-	2	3	2	2
(l)	8	9	-	9	9	9	9

Por último, presentamos una tabla comparativa del rendimiento obtenido en el presente trabajo y resultados publicados considerados representativos (ver Tablas 7.10 y 7.11). Los resultados para CENPARMI son comparables con los publicados, aunque deseables de mejorar. Esta es una base de datos de difícil clasificación. Por ejemplo, Suen [97] obtiene un 98.85 % de reconocimiento entrenando redes neuronales con 450.000 muestras [59]. Para MNIST, el clasificador presentado ha dado muy buenos resultados. Observamos que los trabajos que han obtenido mayor rendimiento en cuanto a porcentajes utilizan un conjunto de entrenamiento modificado, con mayor cantidad de imágenes que surgen de aplicar deformaciones a los patrones del conjunto de entrenamiento. Según [98] los mejores resultados para MNIST se han obtenido con esta técnica, con lo cual se podría considerar de incorporarla para nuestros sistemas como un trabajo a futuro. Además, algunos de estos trabajos utilizan una estructura de clasificador muy compleja como en el caso de la red Convolucional, o inclusive de un MLP con una cantidad de pesos o sinapsis del orden de los 12 millones, lo cual obliga a utilizar tarjetas gráficas para entrenar la red durante varias horas [98][99]. Más allá de las diferencias en la salida, en cuanto al tratamiento de patrones ambiguos (cuestión que en general los otros sistemas no realizan), el rendimiento obtenido por los sistemas SCLAM es alto, sobre todo teniendo en cuenta la gran disminución de costo computacional debido principalmente a la alta reducción del tamaño de los descriptores considerados (superior o igual al 75 %), y de la utilización de clasificadores clásicos con una estructura relativamente sencilla.

Tabla 7.10: Comparación del rendimiento de distintos clasificadores sobre *CENPARMI*.

Método	% Error	Año
GMM [100]	0.65	2011
SVC-RBF [101]	0.85	2004
SVC-RBF [32]	1.10	2002
NN [97]	1.15	1999
KM + SVM [102]	2.40	2004
TWPCA + SCLAM (método propuesto)	*2.60*	*2011*
TW + MCNN [38]	5.30	2000
Combinación SOMs [49]	5.50	2007
TW + MLP [39]	7.80	2003

Tabla 7.11: Comparación del rendimiento de distintos clasificadores sobre *MNIST*.

Método	% Error	Año
Conjunto de MLPs [98][99]	0.31	2011
LCNN + deformaciones elásticas [103]	0.39	2006
CNN + deformaciones elásticas [104]	0.40	2003
TFE-SVM + deformaciones elásticas [45]	0.56	2007
TWPCA + SCLAM (método propuesto)	*0.68*	*2011*
LeNet5 + deformaciones elásticas [45]	0.72	2007
NN+SVM [34]	0.83	2004
TFE-SVM [45]	0.83	2007
Combinación-SVM [95]	0.89	2009
LeNet5 [4]	0.95	1998
Boost.Strumps Haar [105]	1.26	2009
Combinación MLPs [40]	1.40	2004
Distancia Bhattacharyya [106]	1.80	2008

Capítulo 8

Conclusiones y Trabajos Futuros

En este trabajo hemos presentado un sistema para el reconocimiento de patrones que detecta ambigüedades, basado en una estrategia Bayesiana orientada a definir la salida del sistema. Como elementos componentes del reconocedor, en una primera capa o nivel, se utilizaron clasificadores relativamente sencillos y bien posicionados para el problema a tratar. En una segunda capa, la utilización de la Tabla de Confiabilidad estimando cuán confiable es la respuesta de cada clasificador individual frente a un patrón de entrada, permitió decidir cuándo un patrón debía ser considerado bien definido o ambiguo, y en este último caso con qué clases podría confundirse. La Tabla se utilizó también para definir y aplicar estrategias de selección de clasificadores en la etapa de construcción del reconocedor.

La estrategia Bayesiana de combinación de clasificadores que permite detectar ambigüedades y resolverlas constituye la propuesta original de este trabajo de tesis y la principal contribución al estado del arte. Un aspecto que nos interesa destacar a nivel de análisis de la salida, es que la estrategia presentada detecta los patrones ambiguos, indica con qué clases podrían confundirse, y no rechaza patrones de entrada (aunque permite incorporar esta característica), es decir, siempre asocia la entrada con la clase más cercana, indicando si el patrón es considerado ambiguo o no. Esta es una diferencia con las otras estrategias que indican el rechazo de patrones y no distinguen entre patrones bien definidos y los dudosos. Los resultados experimentales muestran que la estrategia Bayesiana supera a las otras estrategias en cuanto a porcentaje de patrones reconocidos para distintos tipos de clasificadores individuales utilizados, disminuyendo el error y aumentando la calidad de la respuesta.

El sistema reconocedor de patrones presentado fue aplicado al problema del reconocimiento de dígitos manuscritos *off-line*, como forma de testear su desempeño. En función de esto, hemos propuesto descriptores basados en características de multirresolución a través del uso de la Transformada Wavelet CDF 9/7 y de Análisis de Componentes Principales, lo que ha permitido mejorar el rendimiento en la clasificación y disminuír el costo computacional.

La experimentación se realizó sobre las bases de datos CENPARMI y MNIST ampliamente referenciadas para este problema. Para la base CENPARMI el mejor resultado fue obtenido con un reconocedor compuesto por cinco clasificadores de tipo Máquinas de Soporte Vectorial (SVM), cada uno dedicado a una característica direccional o global representada por un descriptor basado en la Transformada Wavelet y Análisis de Componentes Principales (TW-PCA). El uso de estos descriptores permitió reducir la dimensión del patrón de entrada en un 75 % con respecto a la imagen inicial. El porcentaje de reconocimiento obtenido por el sistema fue de 97.40 %.

Para la base MNIST el sistema se construyó con cuatro clasificadores en función de la estrategia de selección de clasificadores presentada, dedicados a características direccionales y global, también representadas con descriptores de tipo TW-PCA. En este caso la reducción de la dimensión del patrón de entrada fue de 87.50 %, mientras que el porcentaje de reconocimiento obtenido por el sistema fue de 99.32 %.

Los porcentajes de reconocimiento obtenidos son altos, sobre todo teniendo en cuenta la importante reducción realizada sobre el tamaño del descriptor. Esta reducción impacta fuertemente en los tiempos de entrenamiento de los clasificadores y en el tratamiento de grandes bases de datos como es el caso de MNIST.

Como desafíos a futuro se plantea el estudio de distintos aspectos relacionados con el diseño del sistema reconocedor, como por ejemplo, la incorporación de modificaciones a la estrategia de detección de patrones ambiguos para que permita, además, detectar outliers, es decir, patrones que no pueden asociarse con ninguna clase. El análisis de los errores o patrones mal clasificados, junto con el de los votos y puntajes asignados por el reconocedor en general, podrá ser útil en este punto. Otro aspecto que invita a seguir investigando es la definición de nuevas estrategias para la selección de clasificadores componentes del sistema, y la evaluación del impacto que tiene la utilización de la tabla de confiabilidad en este tema. Por otro lado, creemos que el estudio de la conveniencia de incorporar otro tipo de clasificadores individuales podría aportar a la hora de seguir mejorando la respuesta del sistema.

Apéndices

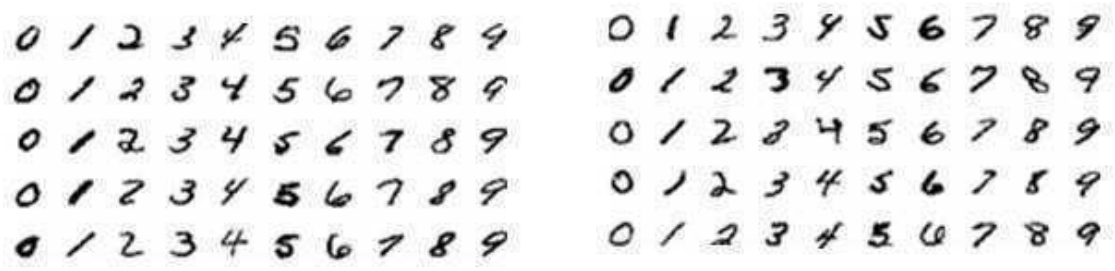
Apéndice A

Bases de Datos

En el presente Anexo se describen las bases de datos de dígitos manuscritos utilizadas para el desarrollo del trabajo: CENPARMI y MNIST. Ambas constituyen un estándar para el entrenamiento y testeo de sistemas reconocedores, lo cual permite comparar los resultados obtenidos con los de la literatura.

A.1. Base de Datos CENPARMI

La base de datos de números manuscritos del *Centre for Pattern Recognition and Machine Intelligence* (CENPARMI) de la Universidad de Concordia, Canadá [3], ha sido utilizada ampliamente en la literatura para medir el rendimiento de los sistemas de clasificación. Está compuesta por imágenes de dígitos escritos a mano sin restricciones provistos por el Servicio Postal de los Estados Unidos, y extraídos de códigos postales manuscritos en los sobres de la correspondencia. Cada número de la base fue digitalizado en dos niveles, en una grilla de 64 x 224 elementos cuadrados de 0.153 mm cada uno, obteniéndose una resolución de aproximadamente 166 pixels por pulgada [3]. La base provee dos conjuntos de patrones rotulados, uno de 'entrenamiento', conformado por 4000 dígitos (400 de cada clase), y otro de 'testeo', conformado por 2000 dígitos (200 por cada clase), este último orientado a evaluar el desempeño de los métodos que se apliquen. Las imágenes de la base fueron normalizadas en tamaño a 16 x 16 píxeles, valor utilizado por varios autores [50] [31] [47]. La Figura A.1 presenta muestras del conjunto de entrenamiento y de testeo, permitiendo observar la característica de 'escritura irrestricta' de las mismas, representada por los diferentes estilos de escritura, inclinaciones y trazos. Una característica importante de esta base de datos es que el conjunto de entrenamiento y el de testeo, incluyen ejemplos que son ambiguos, inclasificables y aún mal clasificados.



(a) Conjunto de Entrenamiento (b) Conjunto de Testeo

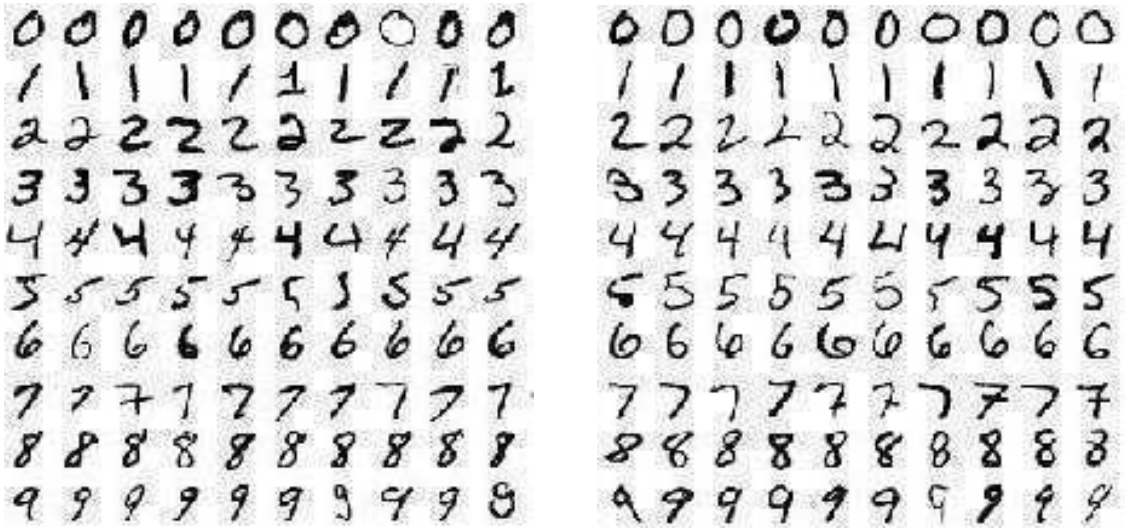
Figura A.1: Dígitos manuscritos de ejemplo de la base de datos CENPARMI, normalizados en tamaño.

A.2. Base de Datos MNIST

La base de datos MNIST [4] constituye un estándar a la hora de testear y comparar métodos de reconocimiento de dígitos manuscritos. La misma contiene 70000 imágenes de números escritos a mano que incluyen diferentes estilos de escritura, extraídos de un conjunto mucho mayor de patrones manuscritos provisto por NIST, *National Institute of Standards and Technology*, organismo que depende del Departamento de Comercio de los Estados Unidos. El nombre de la base proviene de *Modified NIST*.

MNIST presenta dos conjunto de patrones rotulados, uno de entrenamiento constituído por 60000 imágenes, y otro de testeo, de 10000 imágenes. Cada imagen presenta un tamaño de 28x28 píxeles y está representada en escala de grises de 256 valores. Esta base de datos es de acceso libre [5]. Algunos ejemplos pueden verse en la Figura A.2.

Para obtener las imágenes presentadas en la base, los patrones originales en blanco y negro fueron normalizados en tamaño a 20x20 píxeles, preservando su forma. Se obtuvieron imágenes en escala de grises que luego fueron centradas en patrones de tamaño 28x28, según describe [4].



(a) Conjunto de Entrenamiento (b) Conjunto de Testeo

Figura A.2: Dígitos manuscritos de ejemplo de la base de datos MNIST.

Apéndice B

Tablas de Confiabilidad

En el presente Anexo se presentan las Tablas de Confiabilidad asociadas con los clasificadores utilizados en el Capítulo 7 para construir los sistemas reconocedores finales. Las mismas están construidas sobre clasificadores de tipo Máquinas de Soporte Vectorial, cada uno asociado a un determinado preprocesamiento y a un conjunto de datos particular, y expresan cuán confiable es la respuesta dada por cada reconocedor en función de un enfoque probabilístico bayesiano (ver Capítulo 6). Las Tablas de Confiabilidad se pueden observar en la Figura B.1.

Clase	HR	VT	RD	LD	GL
0	0.927	0.950	0.909	**1.000**	**1.000**
1	0.952	0.930	**1.000**	0.930	0.952
2	0.902	0.974	0.929	0.927	0.952
3	0.881	0.974	0.947	0.975	**1.000**
4	0.952	0.950	0.975	0.951	0.976
5	0.972	0.886	0.974	**1.000**	0.976
6	0.975	0.929	0.975	0.952	**1.000**
7	**1.000**	0.950	**1.000**	**1.000**	0.976
8	0.921	0.970	0.900	0.972	**1.000**
9	0.925	0.902	0.974	0.905	0.974

(a) CENPARM1 – LL1

Clase	HR	VT	RD	LD	GL
0	0.975	0.974	0.930	**1.000**	**1.000**
1	0.952	0.930	**1.000**	0.930	0.930
2	0.907	0.974	0.929	0.925	0.952
3	0.902	0.951	0.946	0.927	**1.000**
4	0.952	0.974	0.975	0.975	0.952
5	0.947	0.950	0.950	**1.000**	0.976
6	0.975	0.951	0.975	0.952	**1.000**
7	**1.000**	0.929	**1.000**	**1.000**	**1.000**
8	0.947	0.943	0.897	0.947	0.973
9	0.897	0.881	0.950	0.905	0.973

(b) CENPARM1 – T2

Clase	HR	VT	RD	LD	GL
0	0.975	0.950	0.909	0.974	**1.000**
1	0.952	0.930	**1.000**	0.930	0.952
2	0.950	0.974	0.929	0.929	0.952
3	0.927	0.974	0.947	0.975	**1.000**
4	0.952	0.950	0.975	0.951	0.976
5	0.973	0.886	0.974	**1.000**	0.952
6	0.976	0.929	0.975	0.952	**1.000**
7	**1.000**	0.950	**1.000**	**1.000**	0.976
8	0.900	0.970	0.897	**1.000**	0.974
9	0.923	0.902	0.950	0.902	0.973

(c) CENPARM1 – LL1T2 PCA64

Clase	HR	VT	RD	LD	GL
0	0.987	0.982	0.988	0.985	**0.990**
1	0.985	0.980	0.982	0.987	**0.990**
2	0.977	0.982	0.982	0.982	0.988
3	0.967	0.974	0.980	0.975	**0.992**
4	0.978	0.978	0.982	0.980	**0.990**
5	0.986	**0.992**	0.973	0.985	**0.995**
6	0.982	**0.992**	0.973	0.982	0.988
7	**0.990**	0.980	0.987	0.982	0.982
8	0.956	0.983	0.954	0.977	0.983
9	0.980	0.977	0.973	0.970	**0.995**

(d) MNIST – LL1 PCA98

Clase	HR	VT	RD	LD	GL
0	0.987	0.979	0.988	0.985	**0.992**
1	0.985	0.980	0.982	0.987	**0.990**
2	0.985	0.977	0.987	0.983	0.988
3	0.970	0.977	0.982	0.972	**0.992**
4	0.973	0.978	0.975	0.978	0.987
5	0.985	0.988	0.981	0.983	**0.992**
6	0.982	**0.992**	0.974	0.980	**0.990**
7	0.986	0.980	0.982	0.980	0.985
8	0.953	0.982	0.955	0.978	0.982
9	0.973	0.972	0.964	0.970	**0.995**

(e) MNIST – T2 PCA98

Clase	HR	VT	RD	LD	GL
0	0.987	0.977	**0.990**	0.985	**0.990**
1	0.985	0.980	0.983	0.987	**0.990**
2	0.980	0.977	0.983	0.978	0.988
3	0.972	0.978	0.977	0.975	**0.992**
4	0.977	0.978	0.982	0.982	0.987
5	0.983	**0.990**	0.978	0.985	**0.992**
6	0.982	**0.992**	0.972	0.980	0.988
7	**0.990**	0.980	0.985	0.978	0.982
8	0.954	0.982	0.951	0.975	0.982
9	0.977	0.975	0.970	0.970	**0.995**

(f) MNIST – LL1T2 PCA196

Figura B.1: Tablas de Confiabilidad para distintos sistemas clasificadores basados en SVMs. Junto a cada Tabla se indica con qué base de datos fue entrenado el sistema y el preprocesamiento utilizado. Los valores de mayor confiabilidad están remarcados en rojo.

Bibliografía

[1] S. Watanabe, *Pattern Recognition: Human and Mechanical*. New York: Wiley, 1985.

[2] A. Jain, R. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 4–37, 2000.

[3] C. Suen, C.Nadal, R. Legault, T. Mai, and L. Lam, "Computer recognition of unconstrained handwritten numerals," *Procs IEEE*, vol. 80, no. 7, pp. 1162–1180, 1992.

[4] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[5] Y. LeCun, "The mnist database of handwritten digits," http://yann.lecun.com/exdb/mnist/.

[6] S. Mori, C. Suen, and K. Yamamoto, "Historical review of ocr research and development," *Proceedings of the IEEE*, vol. 80, no. 7, pp. 1029–1058, 1992.

[7] C. Liu and H. Fujisawa, "Classification and learning methods for character recognition: Advances and remaining problems," in *Machine Learning in Document Analysis and Recognition* (H. F. S. Marinai, ed.), pp. 139–161, Springer, 2008.

[8] F. Bortolozzi, A. BrittoJr, L. Oliveira, and M. Morita, "Recent advances in handwritten recognition," in *Document Analysis* (U. P. et al., ed.), pp. 1–31, 2005.

[9] F. Bortolozzi, A. de Souza BrittoJr, L. Oliveira, and M. Morita, "Recent advances in handwritten recognition," *Document Analysis, Editors: Umapada Pal, Swapan Parui, Bidyut Chaudhuri*, pp. 1–30, 2005.

[10] D. Guillevic and C. Y. Suen, "Cursive script recognition applied to the processing of bank cheques," (Montreal, Canada), pp. 11–14, Proc. of 3rd International Conference on Document Analysis and Recognition, 1995.

[11] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification - Second Edition*. John Wiley and Sons, 2001.

[12] J. Schurmann, *Pattern Classification - A unified view of statistical and neural approaches*. Wiley Interscience, 1996.

[13] A. BrittoJr, R. Sabourin, F. Bortolozzi, and C. Y. Suen, "A two-stage hmm based system for recognizing handwritten numeral srings," (Seattle, USA), pp. 396–400, Proc. of 6th International Conference on Document Analysis and Recognition, 2001.

[14] P.Nicholl, A. Amira, D. Bouchaffra, and R. Perrott, "A statistical multirresolution approach for face recognition using structural hidden markov models," *EURASIP Journal on Advances in Signal Processing*, vol. 2008, pp. 1–13, 2008.

[15] V. Vapnik, *The Nature of Statistical Learning Theory*. Springer Verlag, New York, 1995.

[16] L. Oliveira and R. Sabourin, "Support vector machines for handwritten numerical string recognition," (Washington DC), pp. 39–44, 9th IEEE International Workshop on Frontiers in Handwritten Recognition, IEEE Computer Society, 2004.

[17] T. Pavlidis, *Structural Pattern Recognition*. New York: Springer-Verlag, 1977.

[18] L. Perlovsky, "Conundrum of combinatorial complexity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 6, pp. 666–670, 1998.

[19] C. Bishop, *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[20] R. Lopez and E. Oñate, "A software model for the multilayer perceptron," (Salamanca, Spain), pp. 464–468, IADIS International Conference: Applied Computing, IADIS Press, 2007.

[21] D. Zhang, X. Bai, and K. Cai, "Extended neuro-fuzzy models of multilayer perceptrons," *Fuzzy Sets and Systems*, vol. 142, no. 2, pp. 221–242, 2004.

[22] D. Keysers, "Comparison and combination of state-of-the-art techniques for handwritten character recognition: Topping the mnist benchmark," Technical Report, IUPR Research Group, DFKI and Technical University of Kaiserslautern, 2006.

[23] T. Kohonen, *Self-Organizing Maps*. Springer-Verlag, 2001.

[24] S. Haykin, *Neural Networks A Comprehensive Foundation*. Prentice Hall, 1999.

[25] S. Behnke, M. Pfister, and R. Rojas, "A study on the combination of classifiers for handwritten digit recognition," (Magdeburg/Germany), 3rd International Workshop on Neural Networks in Applications NN '98, 1998.

[26] L. Xu, A. Krzyzak, and C. Suen, "Methods of combining multiple classifiers and their applications to handwritten recognition," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 22, no. 3, pp. 418–435, 1992.

[27] J. Kittler, "Combining classifiers: A theoretical framework," *Pattern Analysis & Applications*, vol. 1, no. 1, pp. 18–27, 1998.

[28] L. Oliveira, R. Sabourin, F. Bortolozzi, and C. Y. Suen, "Automatic recognition of handwritten numerical strings: A recognition and verification strategy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 11, pp. 1438–1454, 2002.

[29] W. Pratt, *Digital Image Processing*. Wiley, New York, 1978.

[30] S. Lee, "Off-line recognition of totally unconstrained handwritten numerals using multilayer cluster neural network," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 6, pp. 648–652, 1996.

[31] S. B. Cho, "Self-organizing map with dynamical node splitting: Application to handwritten digit recognition," *Neural Computation*, vol. 9, pp. 1345–1355, 1997.

[32] C. Liu, K.Nakashima, H. Sako, and H. Fujisawa, "Handwritten digit recognition using state-of-the-art techniques," (Washington, DC, USA), pp. 320–327, IWFHR'02 Proceedings of the Eighth International Workshop on Frontiers in Handwritten Recognition (IWFHR'02), IEEE Computer Society, 2002.

[33] C. Liu, K.Nakashima, H. Sako, and H. Fujisawa, "Handwritten digit recognition: benchmarking of state-of-the-art techniques," *Pattern Recognition*, vol. 36, pp. 2271–2285, 2003.

[34] D. Gorgevik and D. Cakmakov, "An efficient three-stage classifier for handwritten digit recognition," vol. 4, pp. 507–510, 17th International Conference on Pattern Recognition, IEEE Computer Society, 2004.

[35] I. Daubechies, *Ten Lectures on Wavelets*. Soc. Indus. Appl. Math., 1992.

[36] S. Mallat, *A Wavelet Tour of Signal Processing*. Academic Press, 1999.

[37] P. Wunsch and A. Laine, "Wavelet descriptors for multiresolution recognition of handprinted characters," *Pattern Recognition*, vol. 28, no. 8, pp. 1237–1249, 1995.

[38] S. Correia and J. de Carvalho, "Optimizing the recognition rates of unconstrained handwritten numerals using biorthogonal spline wavelets," (Barcelona, Spain), pp. 251–254, Pattern Recognition, 2000. Proceedings. 15th International Conference on, ICPR 2000, 2000.

[39] G. Y. Chen, T. D. Bui, and A. Krzyzak, "Contour-based handwritten numeral recognition using multiwavelets and neural networks," *Pattern Recognition*, vol. 36, no. 7, pp. 1597–1604, 2003.

[40] U. Bhattacharya, S. Vajda, A. Mallick, B. Chaudhuri, and A. Belaid, "On the choice of training set, architecture and combination rule of multiple mlp classifiers for multirresolution recognition of handwritten characters," 9th IEEE Intl. Workshop on Frontiers in Handwriting Recognition, 2004.

[41] C. Chen, C. Chen, and C. Chen, "A comparison of texture features based on svm and som," (Hong Kong), pp. 630–633, 18th International Conference on Pattern Recognition, ICPR 2006, 2006.

[42] D. Romero, A. Ruedin, and L. Seijas, "Wavelet based feature extraction for handwritten numerals," *Image Analysis and Processing (ICIAP 2009), LNCS 5716, Springer*, pp. 374–383, 2009.

[43] R. Ebrahimpour and S. Hamedi, "Handwritten digit recognition by multiple classifier fusion based on decision templates approach," *World Academy of Science, Engineering and Technology*, vol. 57, pp. 560–565, 2009.

[44] B. Zhang, M. Fu, and H. Yan, "A nonlinear neural network model of mixture of local principal component analysis: application to handwritten digits recognition," *Pattern Recognition*, vol. 34, no. 2, pp. 203–214, 2001.

[45] F. Lauer, C. Suen, and G. Bloch, "A trainable feature extractor for handwritten digit recognition," *Pattern Recognition*, vol. 40, pp. 1816–1824, 2007.

[46] C. Suen and J. Tan, "Analysis of errors of handwritten digits made by a multitude of classifiers," *Pattern Recognition Letters*, vol. 26, pp. 369–379, 2005.

[47] S. Lee, "Multilayer cluster neural network for totally unconstrained handwritten numeral recognition," *Neural Networks*, vol. 8, no. 5, pp. 783–792, 1995.

[48] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological Cybernetics*, vol. 43, pp. 59–69, 1982.

[49] L. Seijas and E. Segura, "Detection of ambiguous patterns in a som based recognition system: application to handwritten numeral classification," (Germany), 6th International Workshop on Self-Organizing Maps, Bielefeld University, 2007.

[50] S. B. Cho, "Ensemble of structure-adaptive selforganizing maps for high performance classification," *Information Sciences*, vol. 123, pp. 103–114, 2000.

[51] C. Chou, C. Lin, Y. Liu, and F. Chang, "A prototype classification method and its use in a hybrid solution for multiclass pattern recognition," *Pattern Recognition*, vol. 39, no. 4, pp. 624–634, 2006.

[52] L. Lam and C. Suen, "A theoretical analysis of the application of majority voting to pattern recognition," *Pattern Recognition*, vol. 2, pp. 418–420, 1994.

[53] L. Lam and C. Suen, "Application of majority voting to pattern recognition: An analysis of its behavior and performance," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 27, no. 5, pp. 553–568, 1997.

[54] A. Rahman, H. Alam, and M. Fairhurst, "Multiple classifier combination for character recognition: Revisiting the majority voting system and its variations," (Princeton, NJ, USA), Document Analysis Systems V - 5th International Workshop DAS 2002 - Lecture Notes in Computer Science 2423, D. Lopresti, J. Hu, R. Kashi (Eds.) - IAPR - Springer, 2002.

[55] A. Rahman and M. Fairhurst, "Multiple classifier decision combination strategies for character recognition: A review," *International Journal on Document Analysis and Recognition IJDAR*, vol. 5, pp. 166–194, 2003.

[56] C. Suen and L. Lam, "Multiple classifier combination methodologies for different output levels," (Berlin Heidelberg), Multiple Classifier Systems MCS 2000 - Lecture Notes in Computer Science 1857, J. Kittler, F. Rolli (Eds.) - Springer-Verlag, 2000.

[57] S. Gunter and H. Bunke, "Off-line cursive handwritten recognition using multiple classifier systems– on the influence of vocabulary, ensemble, and training set size," *Optics and Lasers in Engineering*, vol. 43, pp. 437–454, 2005.

[58] V. Frinken and H. Bunke, "Evaluating retraining rules for semi-supervised learning in neural network based cursive word recognition," pp. 31–35, Proceedings of the 10th Intl. Conference on Document Analysis and Recognition (ICDAR 2009), IEEE Computer Society, 2009.

[59] C. Liu and H. Fujisawa, "Classification and learning for character recognition: Comparison of methods and remaining problems," (Seoul), International Workshop on Neural Networks and Learning in Document Analysis and Recognition, 2005.

[60] C. Liu and H. Fujisawa, "Classification and learning methods for character recognition: Advances and remaining problems," *Machine Learning in Document Analysis and Recognition*, vol. 90, pp. 139–161, 2008.

[61] P. Estevez, M. Tesmer, C. Perez, and J. Zurada, "Normalize mutual information feature selection," *IEEE Transactions on Neural Networks*, vol. 20, no. 2, pp. 189–201, 2009.

[62] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[63] Y. Freund and R. Schapire, "A short introduction to boosting," *Journal of Japanese Society for Artificial Intelligence*, vol. 14, no. 5, pp. 771–780, 1999.

[64] J. Liu and P. Gader, "Neural networks with enhanced outlier rejection ability for off-line handwritten word recognition," *Pattern Recognition*, vol. 35, no. 10, pp. 2061–2071, 2002.

[65] C. Chatelain, L. Heutte, and T. Paquet, "A two-stage outlier rejection strategy for numerical field extraction in handwritten documents," *Pattern Recognition*, vol. 4, pp. 224–227, 2006.

[66] J. Milgram, R. Sabourin, and M. Cheriet, "Two-stage classification system combining model-based and discriminative approaches," *17th International Conference on Pattern Recognition (ICPR'04)*, vol. 1, pp. 152–155, 2004.

[67] H. Takahashi and T. Griffin, "Recognition enhacement by linear tournament verification," Proc. Second International Conference Document Analysis and Recognition, 1993.

[68] B. Ripley, *Pattern Recognition and Neural Networks*. Cambridge University Press, 1996.

[69] J. Hertz, A. Krogh, and R. Palmer, *Introduction to the Theory of Neural Computation*. Santa Fe Institute Editorial Board, 1990.

[70] J. Dong, A. Krzyzak, and C. Suen, "Local learning framework for handwritten character recogntion," *Engineering Applications of Artificial Intelligence*, vol. 15, no. 2, pp. 151–159, 2002.

[71] J. Laaksonen, M. Koskela, S. Laakso, and E. Oja, "Picsomcontent-based image retrieval with self-organizing maps," *Pattern Recognition Letters*, vol. 21, no. 13, pp. 1199–1207, 2000.

[72] P. Koikkalainen and E. Oja, "Self-organizing hierarchical feature maps," *IJCNN International Joint Conference on Neural Networks*, vol. 2, pp. 279–284, 1990.

[73] P. Prentis, "Galsom - colour-based image browsing and retrieval with tree-structured self-organising maps," (Bielefeld, Germany), Proceedings of the 6th Int. Workshop on Self-Organizing Maps (WSOM 2007), 2007.

[74] A. Rauber, D. Merkl, and M. Dittenbach, "The growing hierarchical self-organizing map: Exploratory analysis of high-dimensional data," *IEEE Transactions on Neural Networks*, vol. 13, pp. 1331–1341, 2002.

[75] R. Lawrence, G. Almasi, and H. Rushmeier, "A scalable parallel algorithm for self-organizing maps with applications to sparse data mining problems," *Data Mining and Knowledge Discovery*, vol. 3, pp. 171–195, 1999.

[76] B. Silva and N. Marques, "A hybrid parallel som algorithm for large maps in data-mining," (Guimaraes, Portugal), New Trends in Artificial Intelligence, Associacao Portuguesa para a Inteligencia Artificial (APPIA), J.Neves, M. Santos and J. Machado Editors, 2007.

[77] D. Castro and L. Seijas, "Image retrieval based on text and visual content using neural networks," *Journal of Applied Computer Science Methods (ACSM 2010)*, vol. 2, no. 1, pp. 21–39, 2010.

[78] T. Kohonen, "The self-organizing map," *Proceedings IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.

[79] C. Liu and M.Ñakagawa, "Evaluation of prototype learning algorithms for nearest neighbor classifier in application to handwritten character recognition," *Pattern Recognition*, vol. 34, no. 3, pp. 601–615, 2001.

[80] A. Bellili, M. Gilloux, and P. Gallinari, "An mlp-svm combination architecture for offline handwritten digit recognition. reduction of recognition errors by support vector machines rejection mechanisms," *International Journal on Document Analysis and Recognition*, vol. 5, no. 4, pp. 244–252, 2003.

[81] C. Chang and C. Lin, "Ijcnn 2001 challenge: generalization ability and text decoding," (Washington DC, USA), pp. 1031–1036, Proc. of International Joint Conference on Neural Networks (IJCNN'01), 2001.

[82] C. Burgues, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[83] B. Scholkopf, A. Smola, R. Williamson, and P. Bartlett, "New support vector algorithms," *Neural Computation*, vol. 12, no. 5, pp. 1207–1245, 2000.

[84] P. Chen, C. Lin, and B. Scholkopf, "A tutorial on nu-support vector machines," *Applied Stochastic Models in Business and Industry*, vol. 21, no. 2, pp. 111–136, 2005.

[85] R. Collobert, S. Bengio, and Y. Bengio, "Parallel mixture of svms for very large scale problems," *Neural Computation*, vol. 14, no. 5, pp. 1105–1114, 2002.

[86] J. Dong, A. Krzyzak, and C. Y. Suen, "A practical smo algorithm," (Quebec City, Canada), Proc. of 16th International Conference on Pattern Recognition (ICPR), 2002.

[87] B. Scholkopf, S. Mika, C. Burgues, P. Knirsch, K. Muller, G. Ratsch, and A. Smola, "Input space vs. feature space in kernel-based methods," *IEEE Trans. on Neural Networks*, vol. 10, no. 5, pp. 1000–1017, 1999.

[88] R. Collobert and S. Bengio, "Svmtorch: Support vector machines for large-scale regression problems," *Journal of Machine Learning Research*, vol. 1, pp. 143–160, 2001.

[89] G. Dreyfus, *Neural Networks. Methodology and Applications*. Springer-Verlag Berlin Heidelberg, 2005.

[90] A. Graps, "An introduction to wavelets," *IEEE Computational Science and Engineering*, vol. 2, no. 2, 1995.

[91] P. Shukla, "Complex wavelet transforms and their applications," (Glasgow (United Kingdom)), M.Phil. Thesis, Dept.of Electronic and Electrical Engineering, University of Strathclyde, 2003.

[92] D. Zhang, W. Zhang, and X. Yang, "Perceptually-adaptive in-band preprocessing for 3-d wavelet video coding," *Optical Engineering Letters - SPIE*, vol. 45, no. 3, 2006.

[93] A. Skodras, C. Christopoulos, and T. Ebrahimi, "JPEG2000: The upcoming still image compression standard," *Elsevier, Pattern Recognition Letters*, vol. 22, pp. 1337–1345, 2001.

[94] L. Kaplan and R. Murenzi, "Pose estimation of sar imagery using the two dimensional continuous wavelet transform," *Pattern Recognition Letters*, vol. 24, pp. 2269–2280, 2003.

[95] L. Seijas and E. Segura, "Detection of ambiguous patterns using svms: Application to handwritten numeral recognition," *Computer Analysis of Images and Patterns (CAIP 2009) - Lecture Notes in Computer Science, LNCS 5702*, pp. 840–847, 2009.

[96] Y. Huang and C. Suen, "Combination of multiple experts for the recognition of unconstrained handwritten numerals," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, pp. 90–94, 1995.

[97] C. Suen, K. Kiu, and N. Strathy, "Sorting and recognizing cheques and financial documents," *Document Analysis Systems: Theory and Practice, S.W.Lee and Y.Nakano (Eds.), LNCS 1655, Springer*, pp. 173–187, 1999.

[98] D. Ciresan, U. Meier, L. Gambardella, and J. Schmidhuber, "Handwritten digit recognition with a committee of deep neural nets on gpus," *Technical Report No. IDSIA-03-11. IDSIA / USI-SUPSI, Dalle Molle Institute for Artificial Intelligence, Switzerland*, 2011.

[99] D. Ciresan, U. Meier, L. Gambardella, and J. Schmidhuber, "Deep big simple neural nets for handwritten digit recognition," *Neural Computation*, vol. 22, no. 12, pp. 3207–3220, 2010.

[100] X. Chen, X. Liu, and Y. Jia, "Discriminative structure selection method of gaussian mixture models with application to handwritten digit recognition," *Neurocomputing*, vol. 74, no. 6, pp. 954–961, 2011.

[101] C. Liu, K.Nakashima, H. Sako, and H. Fujisawa, "Handwritten digit recognition: investigation of normalization and feature extraction techniques," *Pattern Recognition*, vol. 37, pp. 265–279, 2004.

[102] F. Chang, C. Chou, C. Lin, and C. Chen, "A prototype classification method and its application to handwritten character recognition," *Pattern Recognition*, vol. 39, no. 4, pp. 624–634, 2004.

[103] M. Ranzato, C. Poultney, S. Chopra, and Y. L. Cun, "Efficient learning of sparse representations with an energy-based model," in *Advances in Neural Information Processing Systems - NIPS 2006* (J. et al., ed.), MIT Press, 2006.

[104] P. Simard, D. Steinkraus, and J. Platt, "Best practices for convolutional neural networks applied to visual document analysis," pp. 958–962, Proceedings of the 7th Intl. Conference on Document Analysis and Recognition (ICDAR 2003), IEEE Computer Society, 2003.

[105] B. Kegl and R. Busa-Fekete, "Boosting products of base classifiers," (Montreal, Canada), Proceedings of the 26th International Conference on Machine Learning, 2009.

[106] Y. Wen and P. Shi, "A novel classifier for handwritten numeral recognition," (Las Vegas), pp. 1321–1324, IEEE Int.Conf. on Acoustics, Speech, and Signal Processing, 2008.